(NASA-CR-153187) EVALUATING AND MINIMIZING NOISE IMPACT DUE TO AIRCRAFT FLYOVER Final Report (Virginia Univ.) 78 p HC A05/MF A01

N80-23874

Unclas
G3/45 18055

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

SCHOOL OF ENGINEERING AND APPLIED SCIENCE

UNIVERSITY OF VIRGINIA

Charlottesville, Virginia 22901

A Final Report

EVALUATING AND MINIMIZING NOISE IMPACT DUE TO AIRCRAFT FLYOVER

Submitted to:

NASA Scientific and Technical Information Facility
P. O. Box 8757
Baltimore/Washington International Airport
Baltimore, Maryland 21240

Submitted by:

Ira D. Jacobson
Professor

Gerald Cook
Professor

Report No. UVA/528166/MAE80/102
May 1980

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

Members of the faculty who teach at the undergraduate and graduate levels and a number of professional engineers and scientists whose primary activity is research generate and conduct the investigations that make up the school's research program. The School of Engineering and Applied Science of the University of Virginia believes that research goes hand in hand with teaching. Early in the development of its graduate training program, the School recognized that men and women engaged in research should be as free as possible of the administrative duties involved in sponsored research. In 1959, therefore, the Research Laboratories for the Engineering Sciences (RLES) was established and assigned the administrative responsibility for such research within the School.

The director of RLES—himself a faculty member and researcher—maintains familiarity with the support requirements of the research under way. He is aided by an Academic Advisory Committee made up of a faculty representative from each academic department of the School. This Committee serves to inform RLES of the needs and perspectives of the research program.

In addition to administrative support, RLES is charged with providing certain technical assistance. Because it is not practical for each department to become self-sufficient in all phases of the supporting technology essential to present-day research, RLES makes services available through the following support groups: Machine Shop, Instrumentation, Facilities Services, Publications (including photographic facilities), and Computer Terminal Maintenance.

A Final Report

EVALUATING AND MINIMIZING NOISE IMPACT DUE TO AIRCRAFT FLYOVER

Submitted to:

NASA Scientific and Technical Information Facility
P. O. Box 8757
Baltimore/Washington International Airport
Baltimore, Maryland 21240

Submitted by:

Ira D. Jacobson
Professor

Gerald Cook
Professor

Department of Mechanical and Aerospace Engineering
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/528166/MAE80/102 Copy No. 1

May 1980

I. INTRODUCTION

Contained in this report are the results of a study on the evaluation and reduction of noise impact to a community due to aircraft landing and takeoff operations. This work is a continuation of the methods and results of a previous study done by the same authors (under NASA Grant NSG-1509, reference 1). For completeness some repetition of the earlier work is included.

The previous work considered only a single aircraft using a single approach/landing trajectory. Models of population distribution, aircraft noise signature, and aircraft flight path were developed, and a suitable annoyance model adopted. A performance index to be minimized was formed from the annoyance model and constraints. The current study has examined the case of multiple aircraft, flying on several trajectories, for either the case of approach/landings or for takeoffs. A superior, more realistic model of the flight path has also been developed. As in the earlier work, the annoyance criterion used is the noise impact index (NII). The algorithm developed has been applied to Patrick Henry International Airport.

Discussions of the various models, the performance index, optimization methods, and results appear in the following sections.

II. PROBLEM FORMULATION

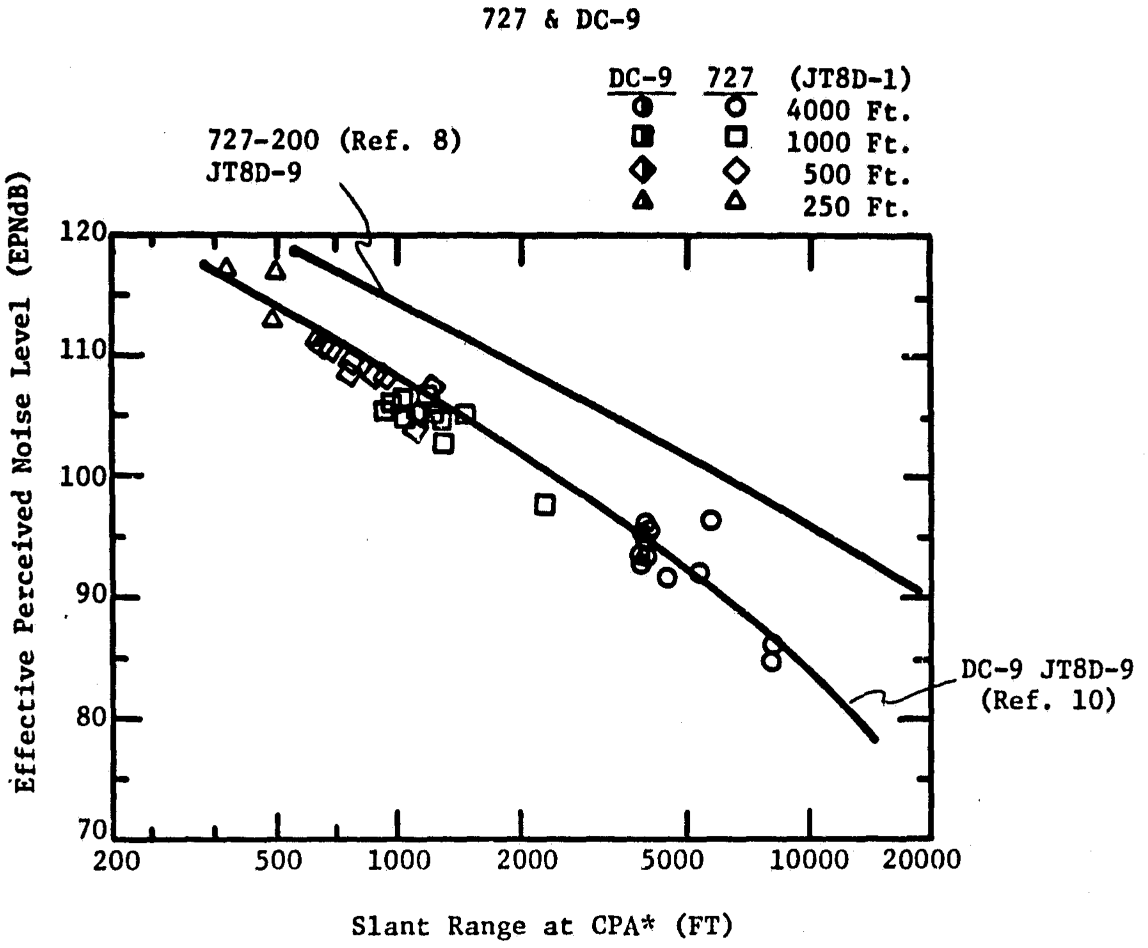
OVERVIEW

The problem considered is that of determining the "best" set of aircraft landing and/or takeoff paths from any airport which minimizes the noise impact on the surrounding community. There are five major aspects of this problem which must be modelled: (1) aircraft noise signatures, (2) population distributions, (3) a cost function or performance index, (4) the aircraft flight paths, and (5) constraints on the aircraft (based upon aircraft dynamics), passenger comfort, safety, and maximum noise exposure for any population group. In addition, a flight path optimization scheme must be adopted. A modular concept has been employed so that any section of the problem may be modified with relative ease. The following sections describe each of these in detail.

A. Aircraft Noise Signatures

An aircraft noise signature gives a description of the noise emanating from an aircraft. Many such representations are available. The one adopted here is a simple model to facilitate computation; however, it can be replaced with more complex and accurate models. One such model is available through the use of the Aircraft Noise Source and Contour Estimation computer programs (see references 2, 3). The aircraft noise signature used in this study is obtained using data from reference 4. Here the effective perceived noise level (EPNdB) is given as a function of slant range to the closest point of approach for a variety of aircraft. A typical plot of the slant range variation for two different aircraft is shown in Figure 1. These data were fit using standard least squares techniques to yield an expression for EPNdB given by

ORIGINAL PAGE IS
OF POOR QUALITY



FLYBY NOISE LEVEL
(1.93 - 1.95 EPR 727 Aircraft) FIG. A-1**
(1.94 EPR DC-9 Aircraft) FIG. D-1**

*Closest Point of Approach
**FAA-RD-71-83 (Ref. 6)

Figure 1. EPNdB vs. Slant Range

$$\text{EPNdB} = 115 - 22.5 \times \left(\frac{\text{slant range in ft.}}{500}\right)$$

This equation is used for calculation of the maximum noise level at each location for a flyover. A typical footprint for a straight-in approach along a 3-degree glide slope is shown in Figure 2. For other aircraft, similar experimental data must be sought in the literature.

B. Population Distribution Model

To model the distribution of population, a map of the community is overlaid with a grid and the population in each grid section is determined. The population distribution within each section is assumed to be uniform. Several grid geometries were examined (see Figure 3). The geometries include: (1) rectangular sections of equal size, (2) rectangular sections whose dimensions increase with distance from the airport runway, and (3) concentric circles divided by several radial lines. The second scheme was chosen since it requires fewer rectangular sections than the first and is somewhat easier to implement than the third. Computer time required for determining the optimum trajectory varies directly with the number of grid sections. This results in the desire to minimize the number of blocks in the grid. Furthermore, since the noise levels decrease with distance from the aircraft and the aircraft has higher altitude when farther from the runway, the need for high resolution of the population density diminishes with distance from the airport. Grid blocks with larger area may then be used when farther away from the airport.

Within a grid section, the population is determined by use of the SITE II system (reference 5), available on the CDC 7600 computer at the

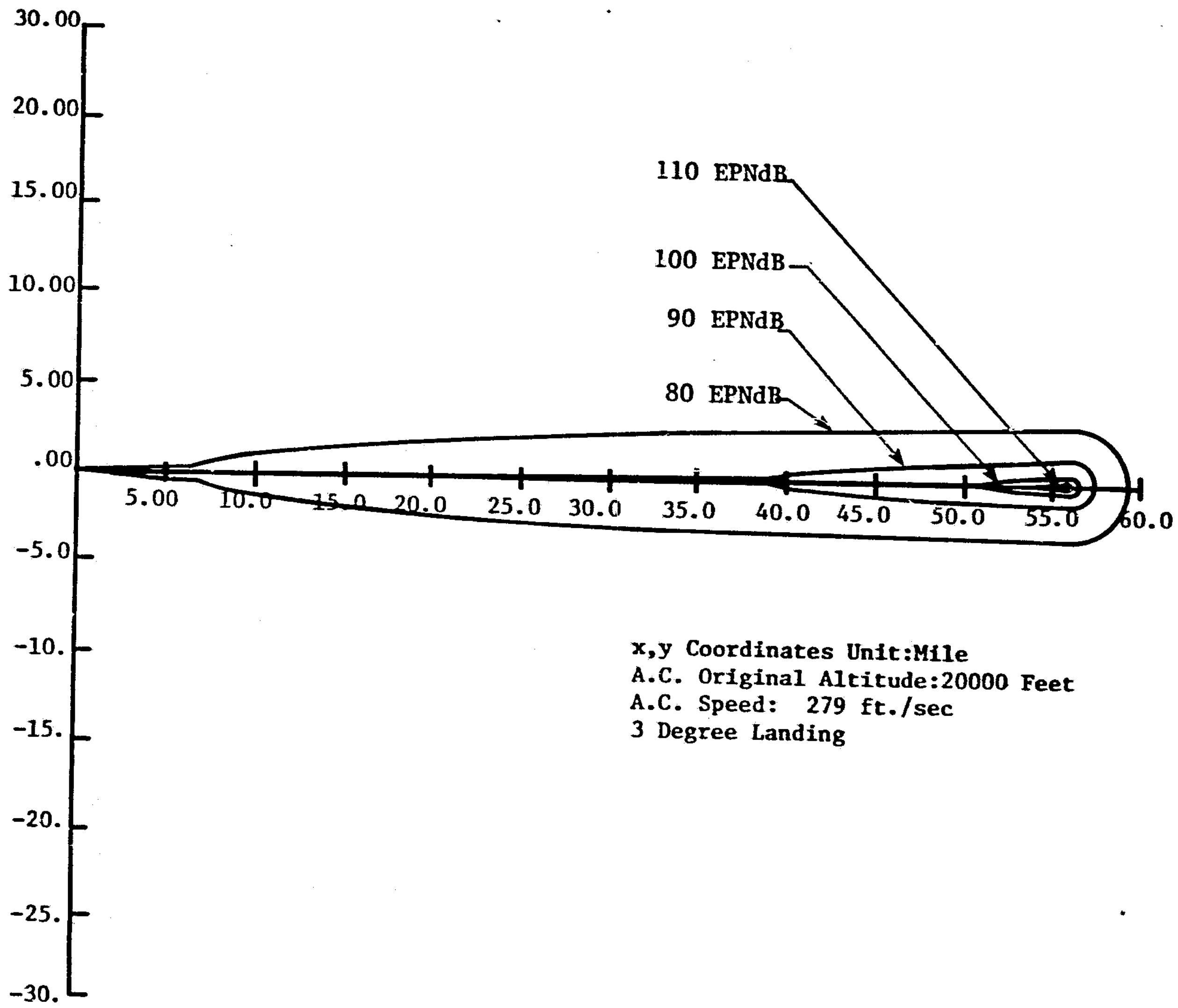
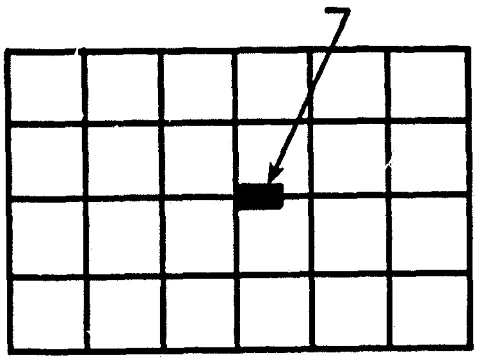
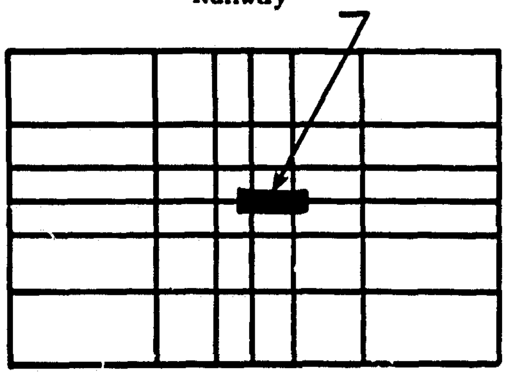


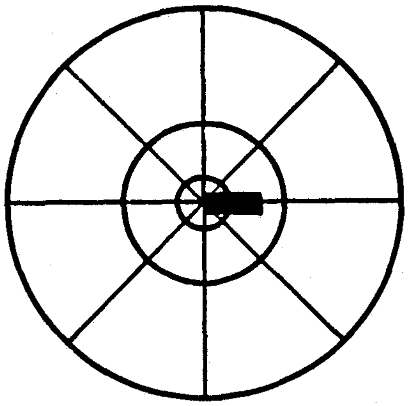
Figure 2. Noise Footprint



1. Equal Size Blocks



2. Variable Size Blocks



3. Concentric Circles

Figure 3. Population Grid Geometry

NASA-Langley facility. This system requires as input the latitude and longitude of a reference point and the coordinates of the corners of each rectangular section. Although SITE II allows for simple retrieval of 1970 census data, there is some question about its resolution capabilities for small grid sections. In addition, in rapidly growing areas the population data may lag the actual population. The SITE II program is capable of producing detailed census information as shown in Figure 4; however, for the present analysis only the population information is used, as indicated.

C. Flight Path Model

There are two ways in which the trajectory of the aircraft may be determined. In one, a discrete time integration of the equations of motion (with control deflections) yields point by point spatial coordinates and orientation. Although this allows the flexibility of explicitly including control constraints as well as dynamic constraints (e.g. maximum roll angle), it requires that a considerable number of states of the system be stored in the optimization routine (i.e. each point of the trajectory in discrete form). In the multi-aircraft, multi-trajectory problem investigated here, such storage requirements are prohibitive.

Thus, another method was adopted which uses only the functional form of the trajectory to describe the flight path. Two possibilities have been investigated: (1) a truncated Fourier series representation and (2) a scheme of line segments joined by smooth arcs.

The Fourier series has the advantage of being able to represent any smooth function over a finite range reasonably well when the series is

ORIGINAL PAGE IS
OF POOR QUALITY

DEMOGRAPHIC PROFILE REPORT PAGE 1

SEVEN CORNERS
SALES TERRITORY
SITE TOTAL

	DEG	MIN	SEC
LATITUDE	38	52	10
LONGITUDE	77	9	20

	1975	1970-1975 CHANGE
POPULATION	369003	-18006
HOUSEHOLDS	138552	1076
PER CAPITA INCOME	$ 7464	$ 2384

ANNUAL COMPOUND GROWTH -0.9%

1970 CENSUS DATA

POPULATION		
TOTAL	387009	100.0%
WHITE	367224	94.9%
NEGRO	15414	4.0%
OTHER	4371	1.1%
SPAN	13839	3.6%

AGE AND SEX	MALE		FEMALE		TOTAL
0-5	19328	10.4%	18646	9.2%	9.8%
6-13	26757	14.5%	25269	12.5%	13.4%
14-17	13645	7.4%	13194	6.5%	6.9%
18-20	7536	4.1%	10413	5.2%	4.6%
21-29	35499	19.2%	39587	19.6%	19.4%
30-39	23840	12.9%	22964	11.4%	12.1%
40-49	23476	12.7%	27719	13.7%	13.2%
50-64	27112	14.7%	30045	14.9%	14.8%
65 +	7859	4.2%	14113	7.0%	5.7%
TOTAL	185052		201950		
MEDIAN(AGE)		27.4		28.6	28.0

FAMILY INCOME (000)		
$0-5	7945	7.8%
$5-7	6942	6.8%
$7-10	14752	14.4%
$10-15	25949	25.4%
$15-25	32623	31.9%
$25-50	12867	12.6%
$50 +	1109	1.1%
TOTAL	102187	
AVERAGE	$15763	
MEDIAN	$14134	

HOME VALUE (000)		
$0-10	339	0.7%
$10-15	1084	2.1%
$15-20	4450	8.6%
$20-25	8491	16.3%
$25-35	17183	33.1%
$35-50	14380	27.7%
$50 +	6012	11.6%
TOTAL	51939	
AVERAGE	$34161	
MEDIAN	$31754	
% OWNER	38.5	

OCCUPATION		
MGR/PROF	68537	41.8%
SALES	12291	7.5%
CLERICAL	48735	29.8%
CRAFT	12810	7.8%
OPERTIVS	6010	3.7%
LABORER	2144	1.3%
FARM	114	0.1%
SERVICE	11469	7.0%
PRIVATE	1663	1.0%

RENT		
$0-100	8737	10.5%
$100-150	35292	42.5%
$150-200	28662	34.5%
$200-250	6645	8.0%
$250 +	3792	4.6%
TOTAL	83128	
AVERAGE	$ 150	
MEDIAN	$ 147	
% RENTER	61.5	

EDUCATION	ADULTS > 25	
0-8	20729	9.6%
9-11	24297	11.3%
12	69170	32.0%
13-15	37764	17.5%
16 +	64003	29.6%

AUTOMOBILES		
NONE	13451	9.8%
ONE	71744	52.2%
TWO	44475	32.3%
THREE+	7872	5.7%

HOUSEHOLD PARAMETERS		
FAM POP	335153	86.6%
INDIVIDS	45881	11.9%
GRP QTRS	5975	1.5%
TOT POP	387009	
NO OF HH'S	137476	
NO OF FAM'S	101961	
AVG HH SIZE	2.8	
AVG FAM SIZE	3.3	

UNITS IN STRUCTURE		
1	66945	48.7%
2	1304	0.9%
3-4	5510	4.0%
5-9	11809	8.6%
10-49	31569	23.0%
50 +	20288	14.7%
MOBILE	125	0.1%

HOUSEHOLDS WITH:		
TV	126239	91.8%
WASHER	71594	52.1%
DRYER	54258	39.5%
DISHWSH	56277	40.9%
AIRCOND	79438	57.8%
FREEZER	28600	20.8%
2 HOMES	2856	2.1%

CACI, INC

Figure 4. Demographic Profile Report from SITE II

truncated after a few terms. However, it is not able to represent functions with slope discontinuities without introducing "waviness" into the approximation. A large number of terms are needed to reduce this effect. The line segment representation does not have either of these features; however, it can approximate very well functions which describe the types of paths aircraft customarily fly.

The first method begins by generating a starting path which goes from the initial trajectory point to the desired runway, ending up with the proper heading, i.e., the aircraft velocity vector is aligned with the runway. This starting trajectory is generated using the following equation: (see Figure 5)

$$y_s(x) = [m_f(x-x_p) + (y_p-y_o)] \exp\left[- C_1 \left(\frac{x - x_f}{x_o - x_f}\right)\right] + y_o$$

For the vertical motion a simple three-degree descent path was assumed.

Next, the first five Fourier sine harmonics are used to introduce deviations from the starting path. The coordinate system is scaled so that each of the sine functions contributes zero deviation at the end points. Therefore, if the starting path satisfies the boundary conditions, then the path with the deviations will also. An exponentially decaying factor is used to eliminate heading deviations at the final point.

With the deviations, the equations for the path become

$$y = \left\{\sum_{i=1}^{N} \alpha_i \sin\left[i\pi \left(\frac{x - x}{x_f - x_o}\right)\right]\right\} \left[1 - \exp\left(\frac{x - x_f}{C_2}\right)\right] + y_s(x)$$

$$z = \left\{\sum_{i=1}^{N} \beta_i \sin\left[i\pi \left(\frac{x - x_o}{x_f - x_o}\right)\right]\right\} \left[1 - \exp\left(\frac{x - x_f}{C_2}\right)\right] + z_s(x)$$

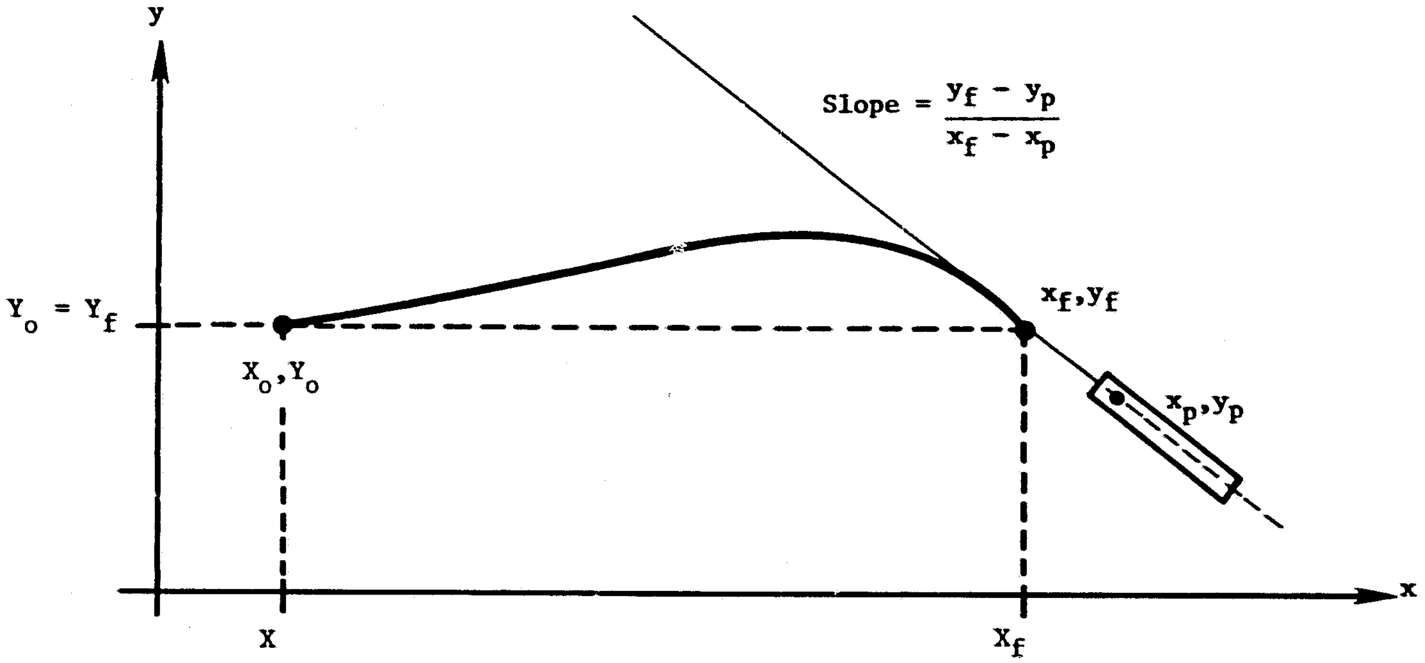


Figure 5. Rotated Coordinate System for Establishing Nominal Flight Trajectory from Initial Point to Runway Approach.

where the α_i and β_i are the unknowns to be determined.

The second flight path model represents the trajectory as a chain of line segments extending from the initial to the final point. Each corner between two segments is "smoothed" with a circular arc whose radius is large enough to insure that the aircraft can perform the turn (see Figure 6). The unknown variables to be determined are the coordinates of the line segment intersections (corner points). For the starting trajectory, the corner points lie equally spaced along a line through each pair of initial and final points. The number of line segments and hence, the number of corner points, per trajectory is determined before the optimization begins. This number is generally small (3 to 5) so that the pilot is not overburdened with required maneuvers.

Both models of the flight path have the advantage of requiring only a small number of parameters to describe the trajectory. This reduces the optimization problem from a variational one to an ordinary one, but care must be taken to see that the various constraints in the problem are met.

D. Constraints

The use of a functional form of the flight path for the trajectory requires the reformulation of constraints into parameters which can be used in the optimization. This is accomplished by translating the steady state solutions of the lateral and longitudinal perturbation equations into geometric constraints. For a detailed derivation of these, see the final report for 1979, Appendix A of reference 1. The constraints are incorporated by determining maximum curvature and slope parameters as a function of aerodynamic and physical constraints.

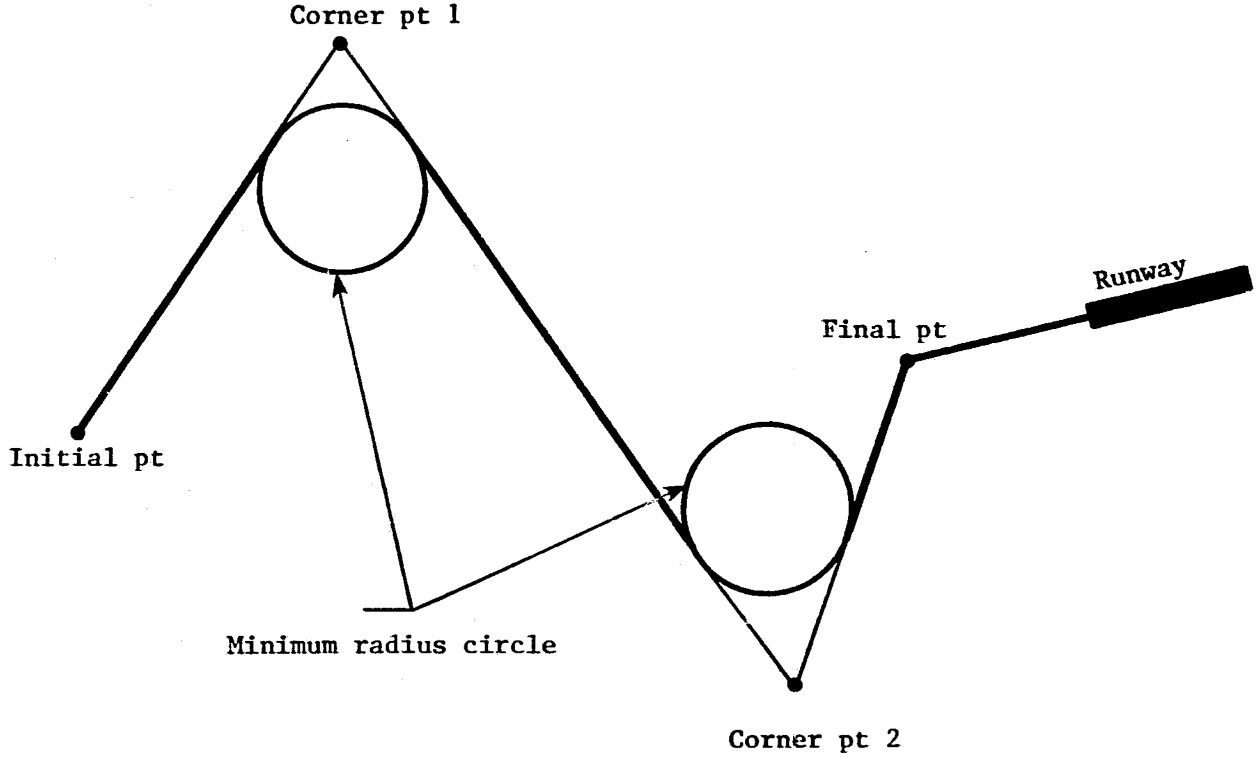


Figure 6. Line Segment Representation of Flight Path

Similar expressions are given in the appendix (referred to above) for constraints on aileron, rudder, and elevator deflections, flight path angle and pitch rate limits.

In addition to the aircraft constraints, there are passenger comfort considerations (e.g. max bank angle), maximum noise exposure levels, and a minimum separation distance between multiple trajectories.

All of the constraints are listed below:

I. Aircraft Dynamic Constraint:

$$\text{Lateral} \quad \left| \frac{\frac{d^2y}{dx^2}}{1 + (\frac{dy}{dx})^2} \right| \leq \frac{C_1 + C_2 C_3}{V_{avg}} \min (\delta r_1, \delta r_2, \delta r_3)$$

where V_{avg} = average velocity of aircraft, C_1, C_2, C_3 are constants for a given aircraft, and the δr's depend upon maximum bank angle and maximum rudder and aileron deflections for a given aircraft.

$$\text{Longitudinal:} \quad \tan \gamma_{c_{max}} < \frac{dz}{dx} < \tan \gamma_{d_{max}}$$

where $\gamma_{c_{max}}$ and $\gamma_{d_{max}}$ are the maximum climb and descent angles.

II. Passenger Comfort Constraint:

$$\left| \frac{[1 + (\frac{dy}{dx})^2]^{3/2}}{\frac{d^2y}{dx^2}} \right| \geq \frac{V^2_{avg}}{C_4 g}$$

where C_4 = 1.9 for 90% passenger satisfaction, 4.5 for 80% satisfaction, g = acceleration due to gravity, and V_{avg} = average velocity of aircraft during the turn.

III. Threshold Noise Constraint

No populated area may receive noise in excess of 95 dB more than N percentage of times per day, where N is a fixed percentage of the number of flights per day. N is made as small as possible for any given case.

IV. Minimum Separation Constraint

A minimum distance of 800 meters (½ mile) must be maintained between any two trajectories at all points (except very close to the runway, where all trajectories must converge).

E. <u>Cost Function</u>

A large number of criteria have been proposed to evaluate noise annoyance (e.g., EPNdB, NNI, sleep interference index, speech interference index, etc.). The recent trend in noise assessment work is toward a universal measure -- the noise impact index (NII). This measure is a weighted day-night model which accounts for population density. It is described in detail in reference 6. Briefly, the total population exposed to each incremental average day-night model sound level is multiplied by the weighting function for that level. The weighting factor $W(L_{dn})$, multiplied by the population exposed to that L_{dn}, is summed and normalized by the total population giving the Noise Impact Index for the area:

$$NII = \frac{\sum_{L_{dn}} P(L_{dn})W(L_{dn})}{\sum_{L_{dn}} P(L_{dn})}$$

A plot of $W(L_{dn})$ appears in Figure 7.

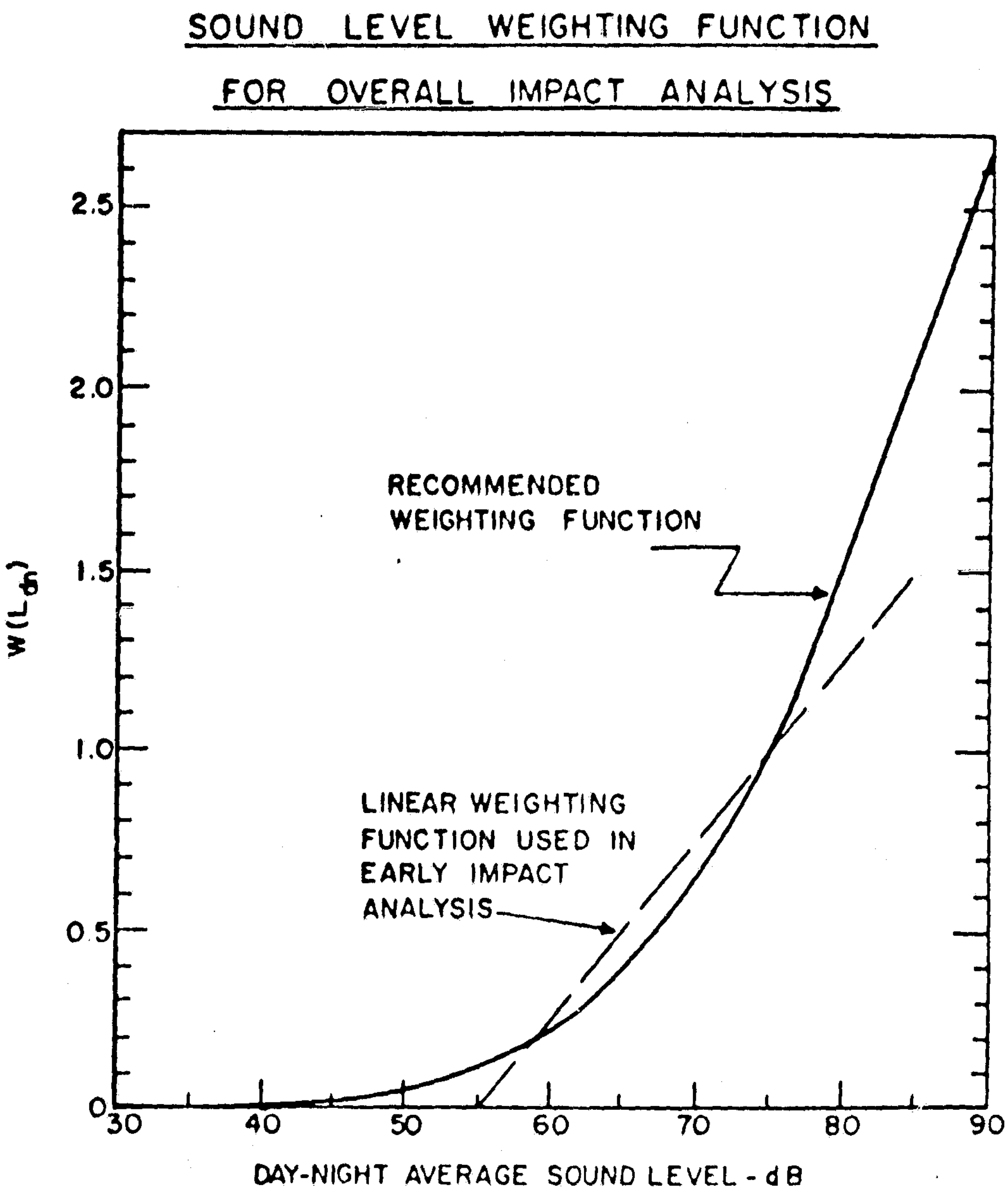


Figure 7. Impact Intensity Weighting Function

The cost function or performance index for the optimization procedure is taken to be the NII plus penalties for violating constraints. Basically, the optimization procedure is set up to "drive" the aircraft trajectories to the path which will minimize the NII and at the same time, not violate any constraints. As an example, the constraint of flight path angle not exceeding a maximum descent angle, γ_d, nor a maximum climb angle, γ_c, is written as

$$\tan \gamma_c < \left| \frac{dz}{dx} \right| < \tan \gamma_d$$

Each is converted to a penalty which is added to the NII in the form

$$\text{Cost} = \text{NII} + K_1P_1 + K_2P_2$$

$$P_1 = \{\max[0, (\tan \gamma_c - dz/dx)]\}^2$$

$$P_2 = \{\max[0, (dz/dx - \tan \gamma_d)]]^2 \qquad K_1, K_2 = \text{constants}$$

As is seen, for values of the flight path angle within the allowable range, no penalty is added; however, for values outside this range, the penalty and thus, the increase in cost, is great. Other penalty terms are added in a like manner.

III. OPTIMIZATION

The optimum set of trajectories is determined by calculating values of the unknowns (the α_i and β_i in the Fourier series model, the corner points in the line segment model) which minimize the total cost (NII plus penalties). Two optimization algorithms have been examined: the method of steepest descent and the Davidon-Fletcher-Powell method. An example of steepest descent is given below. Basically, the method computes the gradient of the cost function, C, with respect to the unknown parameters and then searches along the negative gradient direction for values of the parameters which reduce the cost.

In Figure 8, the point L_1 represents the set of parameters which corresponds to the starting trajectory. The arrow points in the direction of the negative gradient of C (i.e., the direction of decreasing NII). Searching along this direction will yield a new point L_2 which corresponds to a new trajectory with lower NII. The process of computing gradients and searching continues until the cost converges to within a specified tolerance. In this example, the sequence begins at L_1 and converges to L^*, where the NII is an absolute (or global) minimum.

Consider, however, the case where the starting trajectory is characterized by the point Q_1. The optimization process will converge to the point Q^*, which is a relative (or local) minimum. The trajectory characterized by Q^* does give a lower NII than the starting path at Q_1, but the NII at Q^* is still higher than that at L^*.

In this example, it is easily seen that if the starting point lies in Region I, convergence to the global minimum at L^* is assured (likewise for Region II and the convergence to the local minimum at Q^*). The

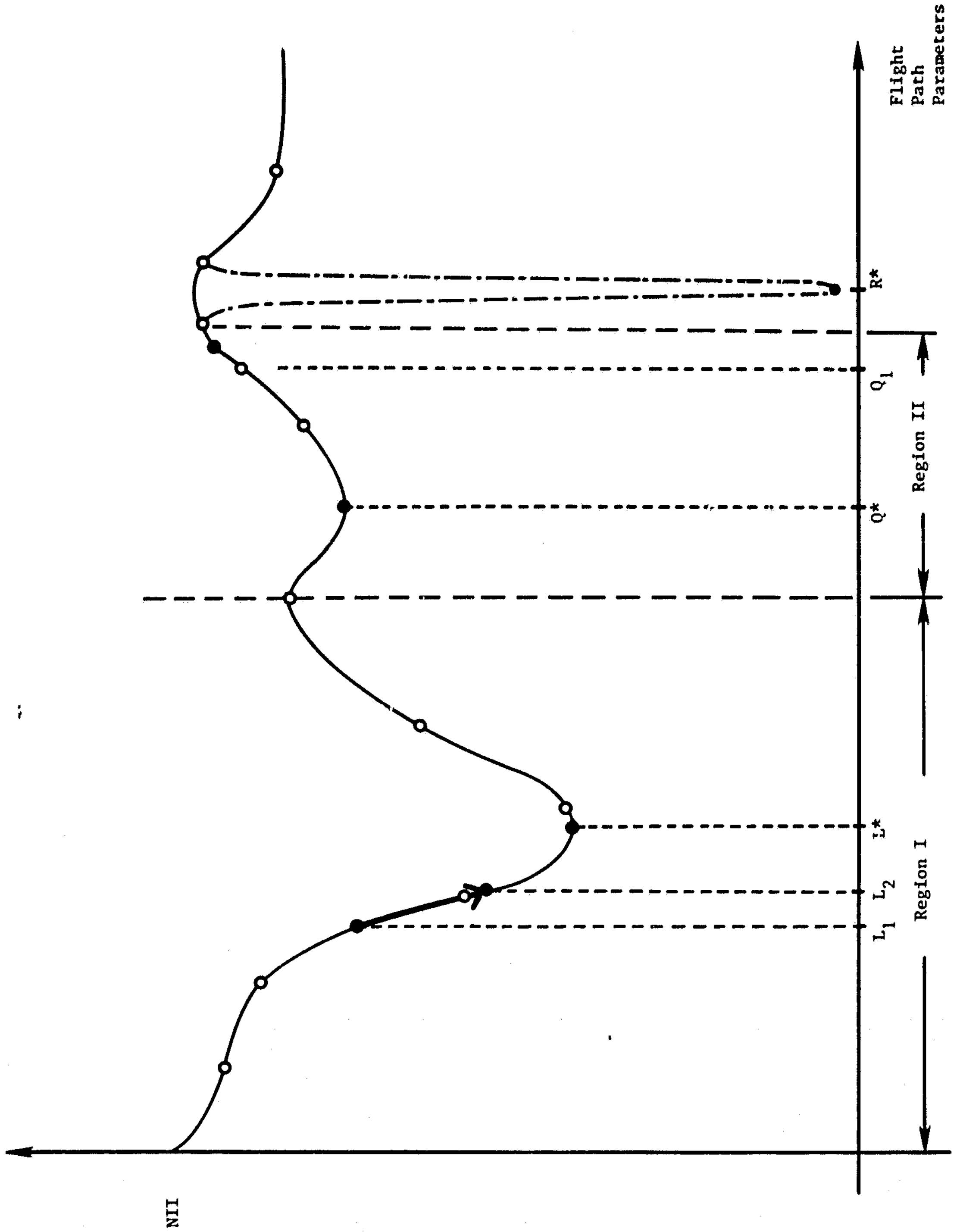


Figure 8. Steepest Descent Example

only way to insure that the point L* is found is to execute the optimization algorithm a number of times with different starting points, as indicated by the open circles in Figure 8. There is the possibility that the cost function has a "sharp" global minimum, such as at the point R*. In such a case, it is likely that none of the starting points chosen would result in convergence to R*. From a practical point of view, though, it is not important that the true global minimum at R* is not found. The range of parameters defining the sharp "well" at R* is so narrow that a pilot could not deviate from the optimal path characterized by R* without greatly increasing the NII. Simply stated, the only optimal path of interest is one whose resulting NII is not overly sensitive to slight variations in the path.

The steepest descent algorithm has the disadvantage of giving slow convergence near the optimal set of unknown parameters; however, significant reduction in the cost (NII) does occur during the first few iterations. A superior algorithm is the Davidon-Fletcher-Powell method, which gives good convergence near the optimum. This method has been employed in this study with satisfactory results. A detailed description of both optimization methods appears in reference 7.

A. The Optimization Algorithm

A computer code has been developed which implements either of the optimization methods described above. Figure 9 shows a flow chart for this code. Initial data (population map, aircraft constraints, initial and final aircraft positions, etc.) are required for each configuration of trajectories and aircraft at a given airport. An initial set of trajectories is either supplied by the user, or a default set is gen-

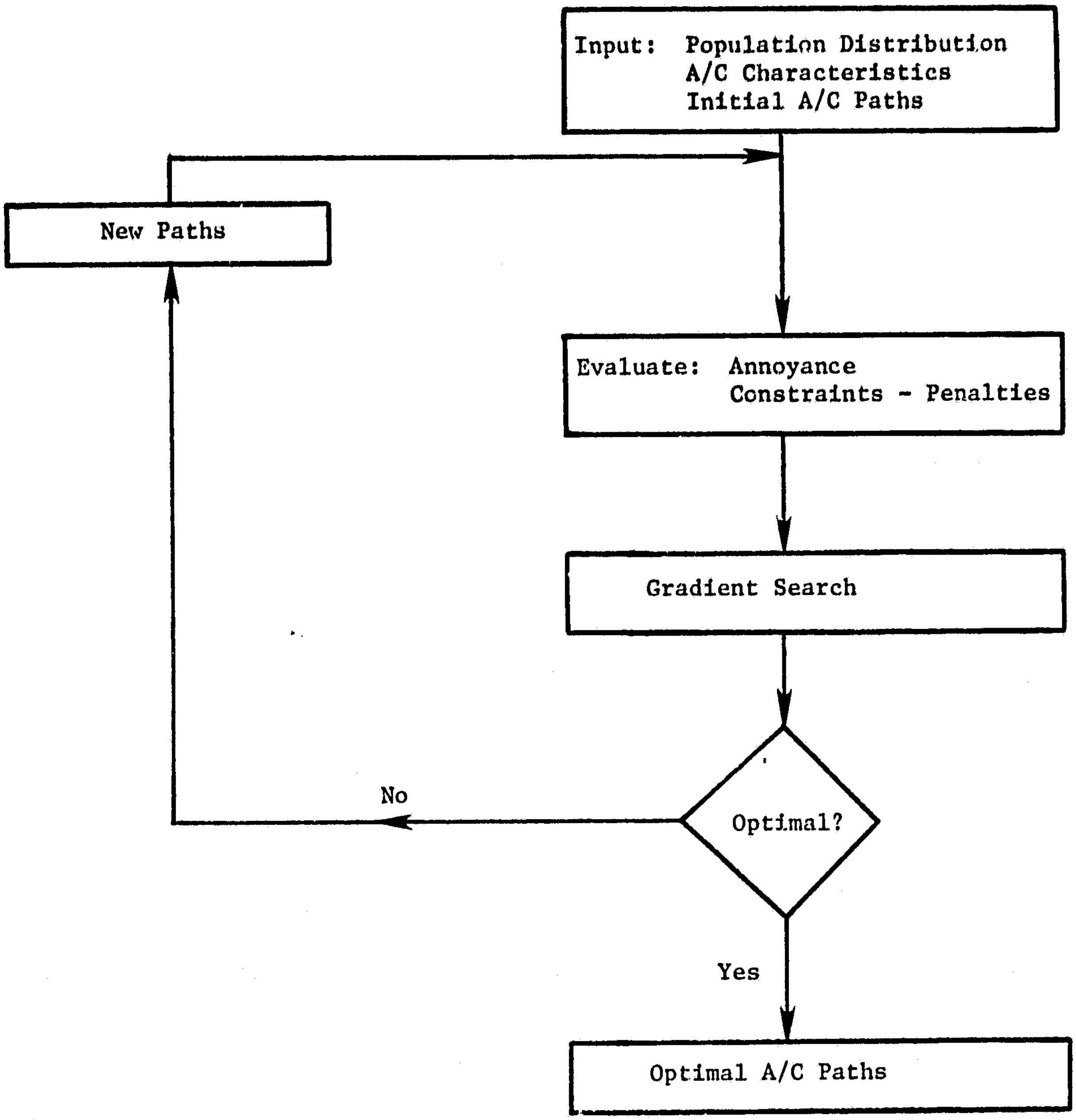


Figure 9. Flow Chart of the Flight Path Optimization Algorithm

erated by the program. The optimization then begins, with successive values of the cost being compared after each iteration. When the difference between successive values is less than a defined stopping criterion, the process terminates.

The code has been written in modular form so that any of the various models (population distribution, cost function, etc.) may be upgraded or modified easily without making major changes in the code. As an example, the noise impact in each population section requires the computation of an integral. While this integral is usually approximated, a more accurate calculation can be made with the simple addition of a subroutine to the program.

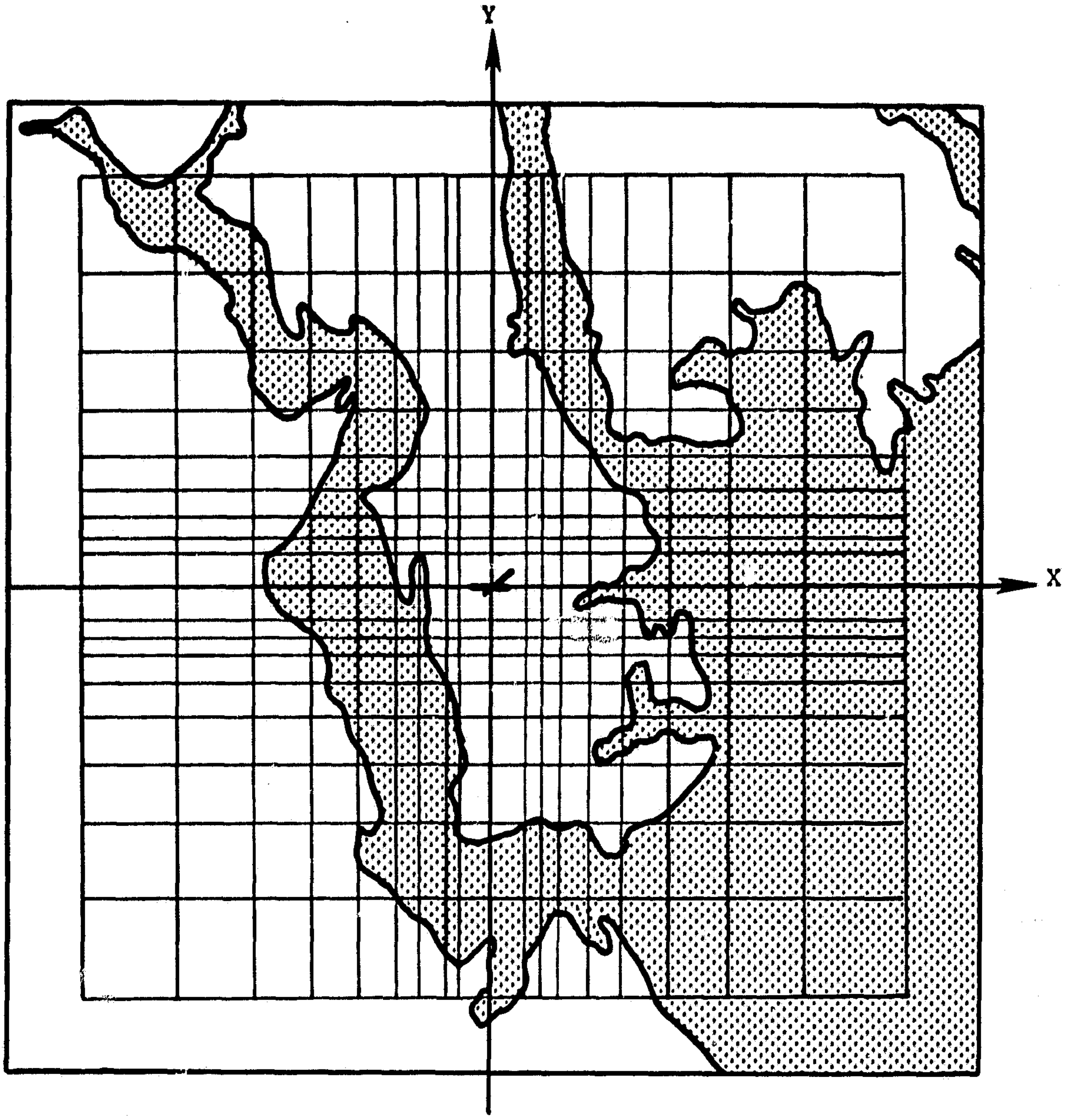
Appendix A contains the FORTRAN code as written for a CDC Cyber 172 machine.

B. Results

All of the cases discussed here involve the Patrick Henry International Airport in Hampton, Virginia. The SITE II program was used to generate the population data for each block as shown in Figure 10. The three entry points referred to, Swing, Franklin, and Cape Charles, are the check points indicated on the ILS approach plate (figure 11).

Reduction in the NII at Patrick Henry Airport is limited by the population distribution. As indicated in Figure 12, most of the people are located in blocks near the runway. During takeoffs and landings, these people will be affected by aircraft noise regardless of the trajectories flown.

(1) The Swing and Franklin entry points are used simultaneously for approach/landings. With the Fourier series model of the flight path,



Population Grid at Patrick-Henry Airport (Partial)

Figure 10. Population Grid Scheme at Patrick-Henry Airport

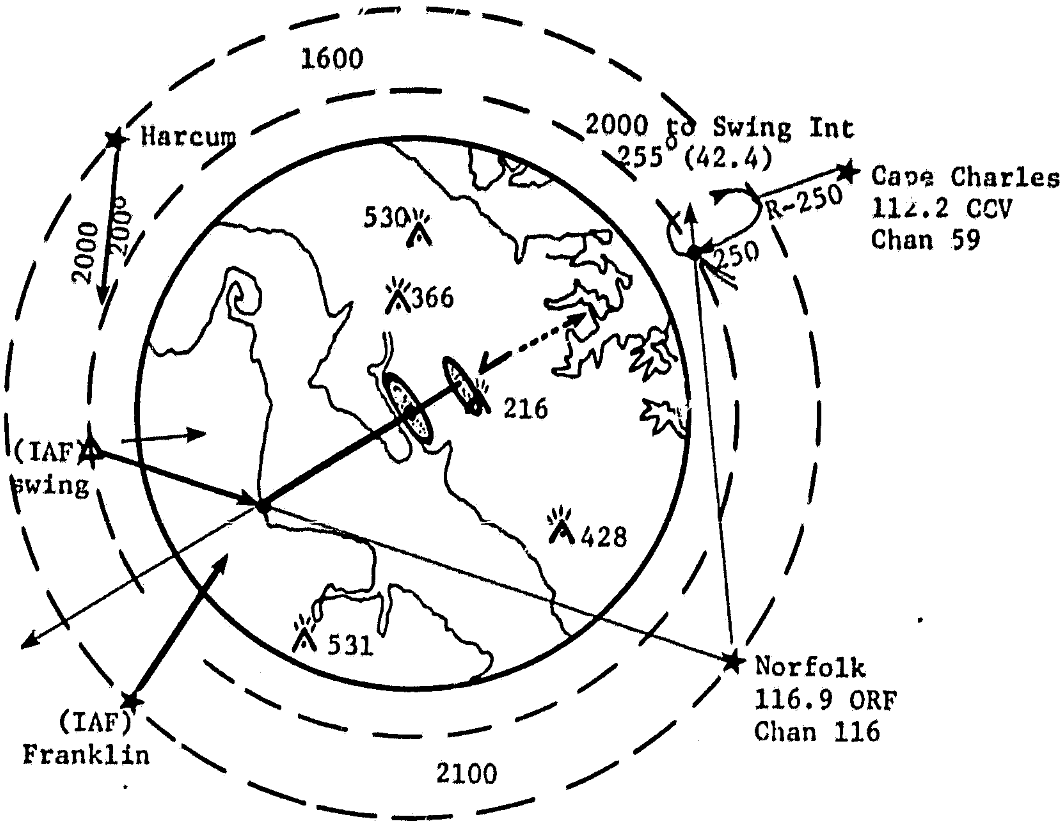


Figure 11. ILS Landing Approach at Patrick Henry Airport

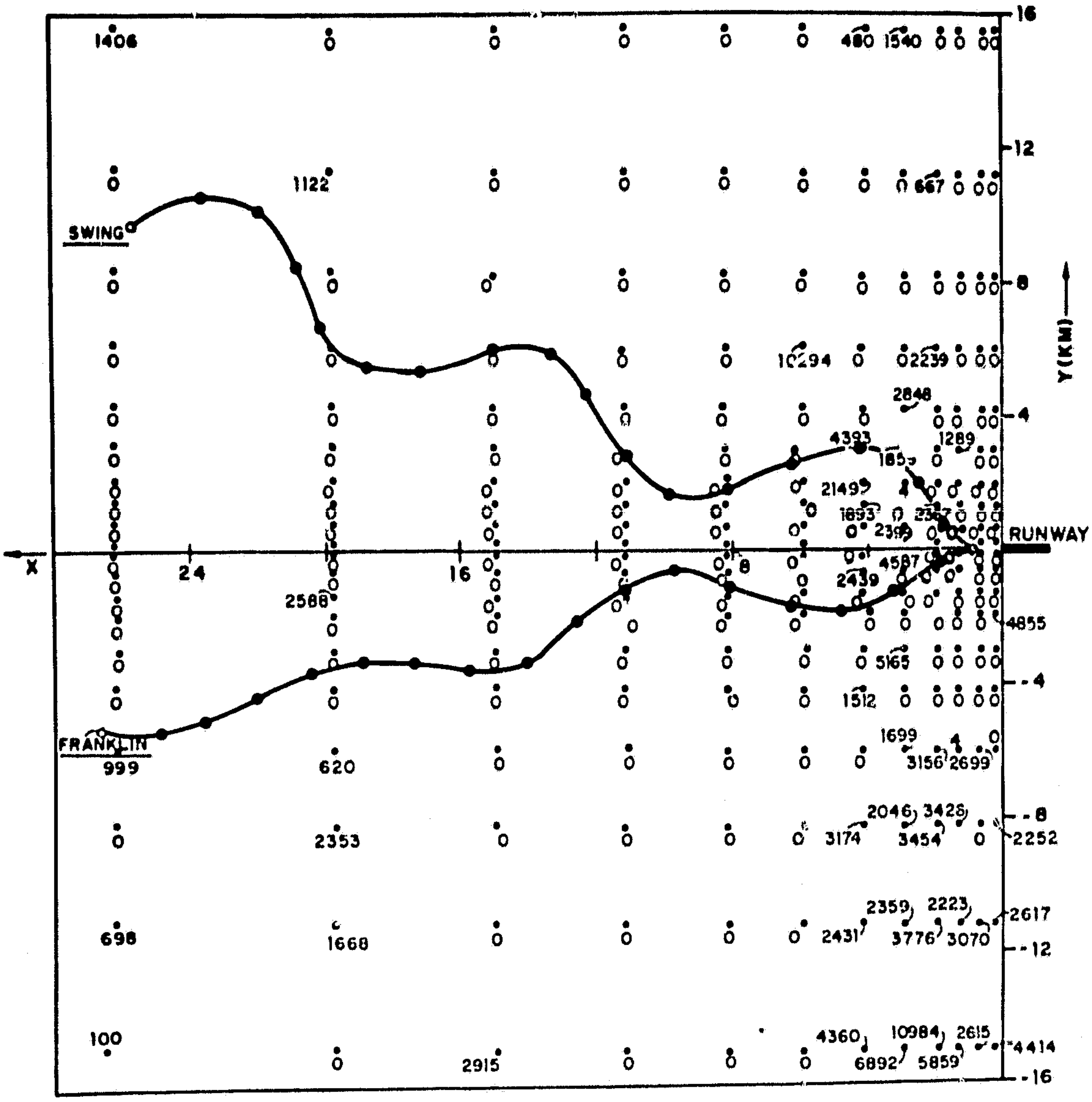


Figure 12. Fourier Series Representation

Table I Sine Harmonic Representation of Flight Paths

(I) Sine harmonic series representation of flight path

Two landing trajectories with entry points at Swing and Franklin, respectively. Aircraft distributions on both of the trajectories are:

	B707	B727
day time	2	2
night time	2	2

Sine harmonic parameter α's			Annoyance	
	Swing entry trajectory	Franklin entry trajectory		
1	5.5074×10^2	-2.8931×10^2	1.5142	Figure 12
2	-7.5166×10^2	5.5448×10^2		
3	1.4734×10^3	-7.7248×10^2		
4	-1.1312×10^3	1.0156×10^3		
5	2.2254×10^3	-1.0825×10^3		

the results obtained are shown in Figure 12. Details appear in Table I. As is easily seen, there is an unnecessary amount of waviness in the trajectories far from the runway. This is caused by the fact that the Fourier series is truncated after five terms. More terms could be included but more computation time would be required. Thus, the line segment model of the flight path has been adopted and is used in all the following cases.

(2) A single trajectory, with one Boeing 707 flying, is determined using the line segment model. The results are shown in Figures 13a and b. Both the Swing and Franklin stations have been used as entry points. There are three segments in each trajectory, requiring only three turning maneuvers from the pilot. This is clearly more realistic than the type of path produced by the Fourier series model. A comparison of the results of the two models shows that the line segment scheme yields slightly higher NII values (3-5% higher than in the Fourier series representation); however, the NII is reduced, compared to existing approach paths, by 4-6%.

(3) Multiple aircraft on multiple trajectories are investigated. Figures 14, 15 and 16 show the results for two, three, and four segments per trajectory. The reduction in NII ranges from 4 to 5%. Details appear in Table II.

(4) The multiple aircraft, multiple trajectory case is repeated (with three segments per trajectory) using Gaussian quadrature to evaluate the integral in the NII computation; a 6% reduction is seen. Figure 17 shows the difference that results when this integration is

ORIGINAL PAGE IS
OF POOR QUALITY

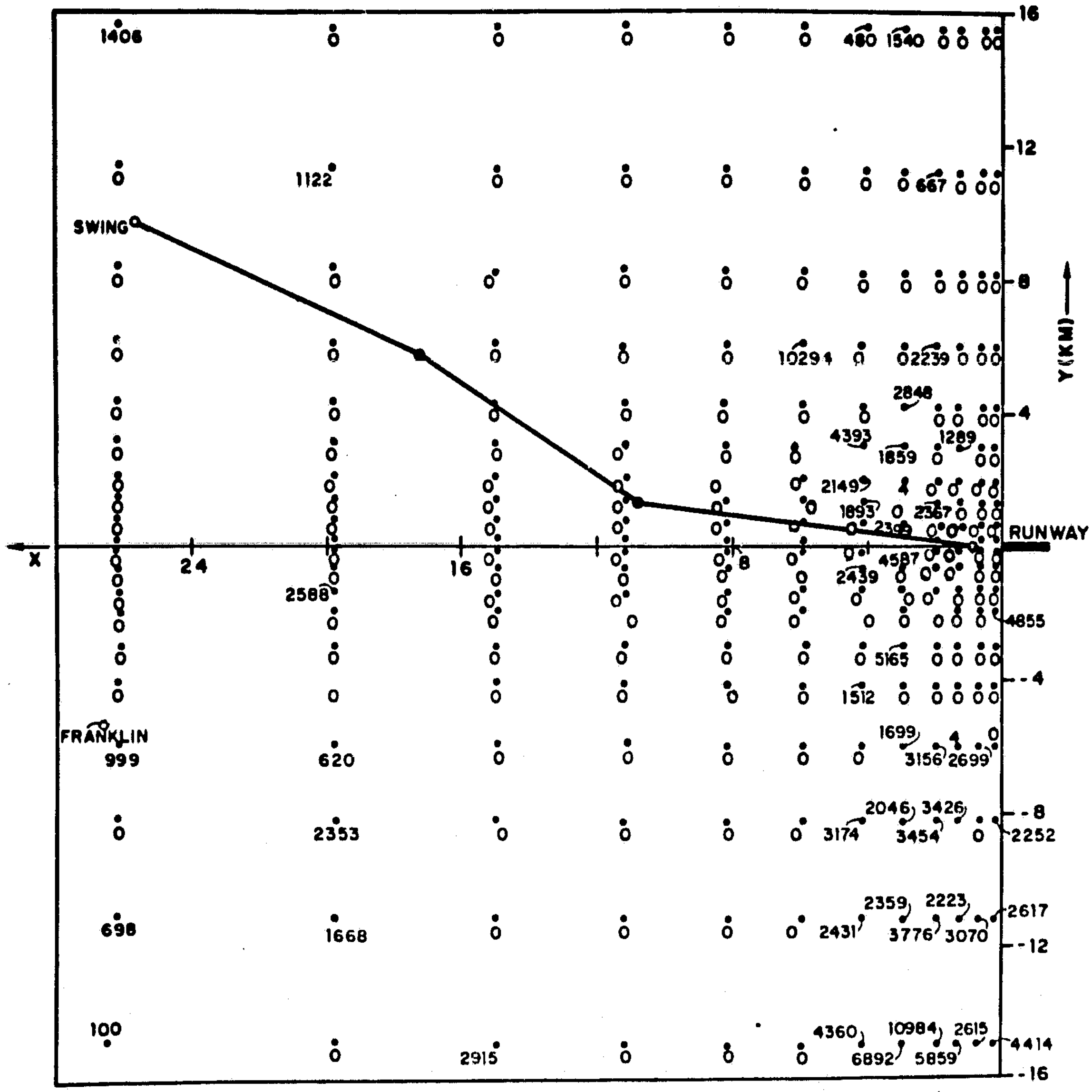


Figure 13a Single Aircraft (Landing)

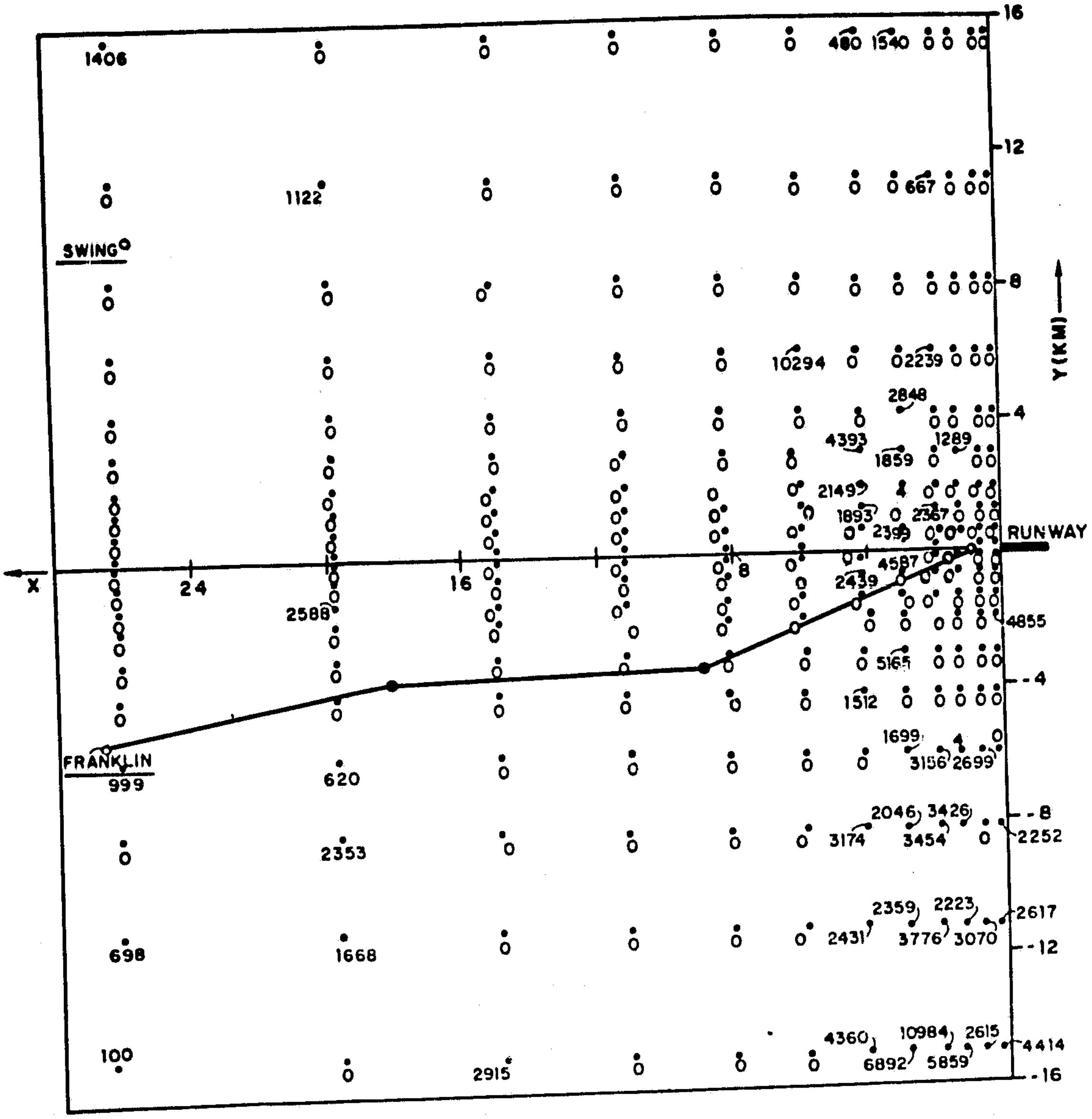


Figure 13b Single Aircraft (Landing)

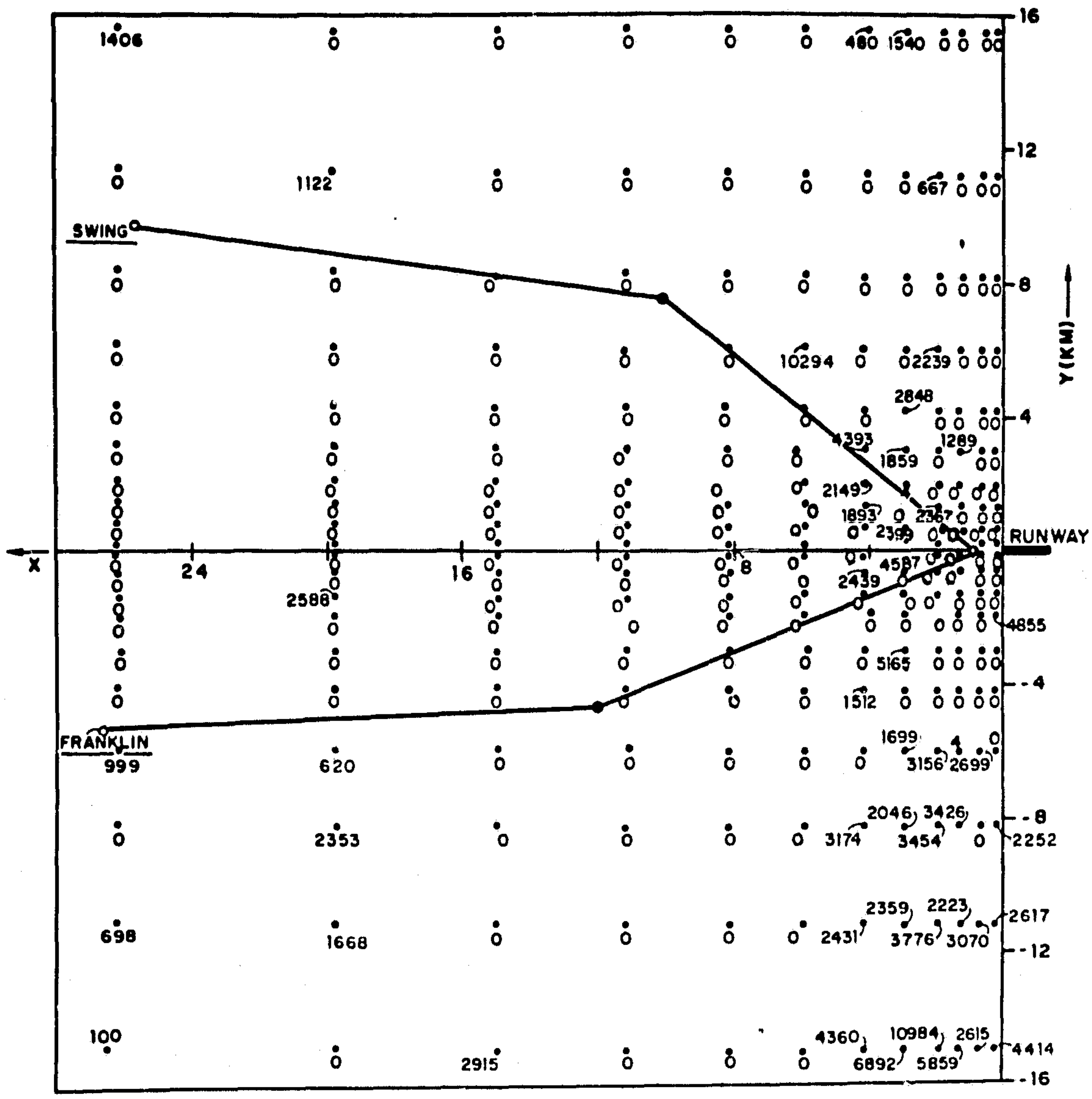


Figure 14a Two Segments per Trajectory (Landing)

ORIGINAL PAGE IS
OF POOR QUALITY

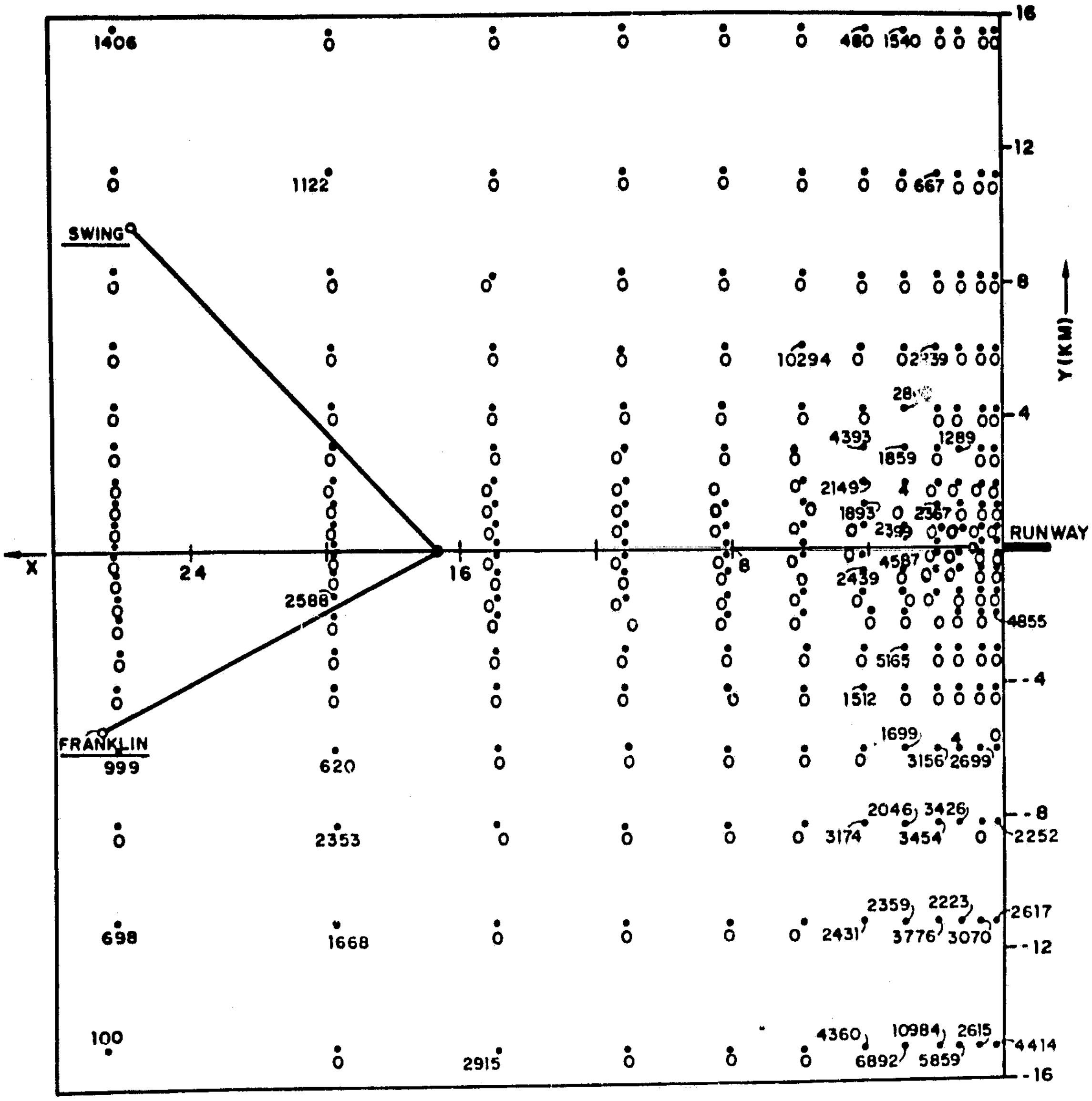


Figure 14b. Existing Landing Approach

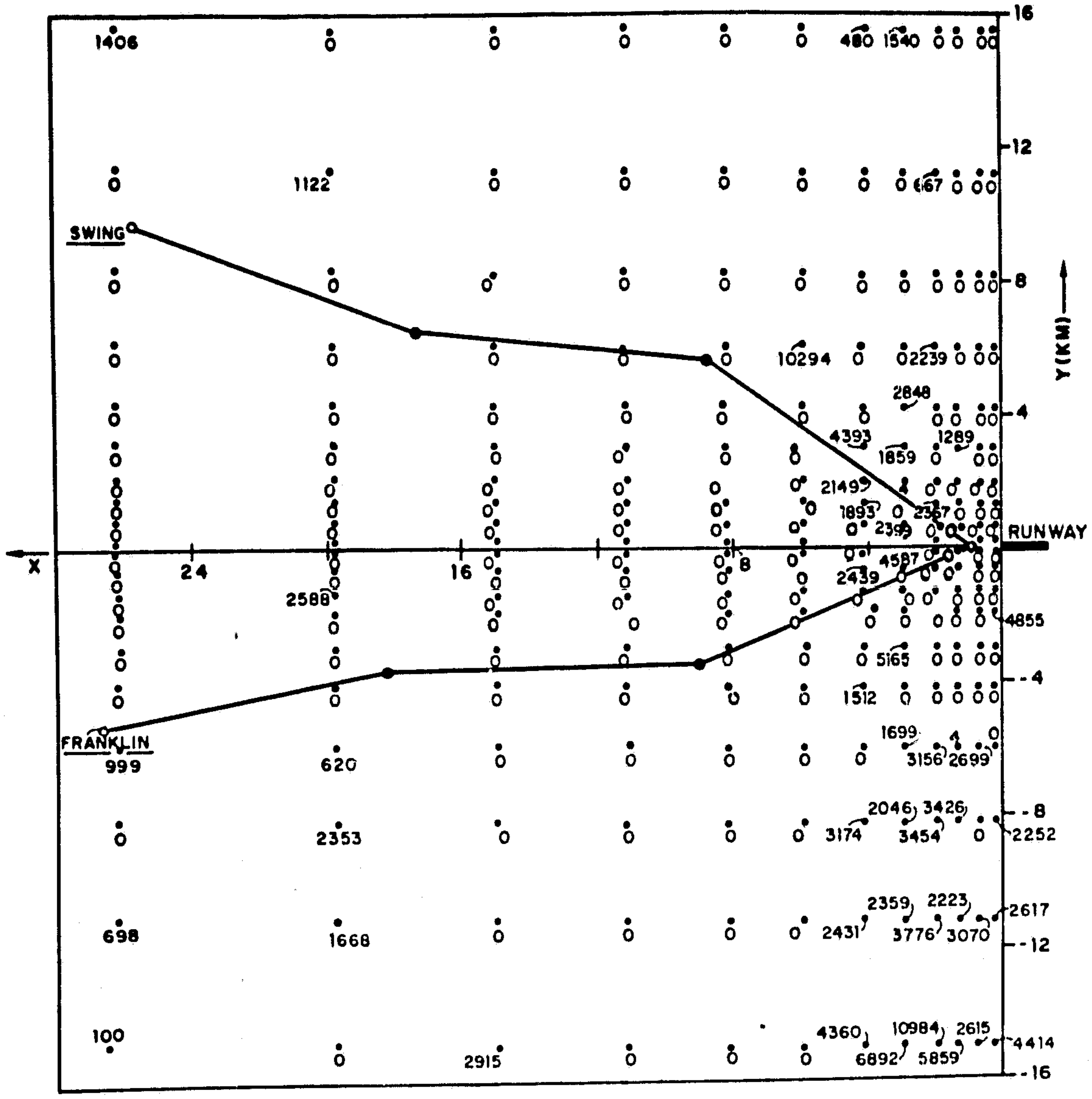


Figure 15. Three Segments per Trajectory (Landing)

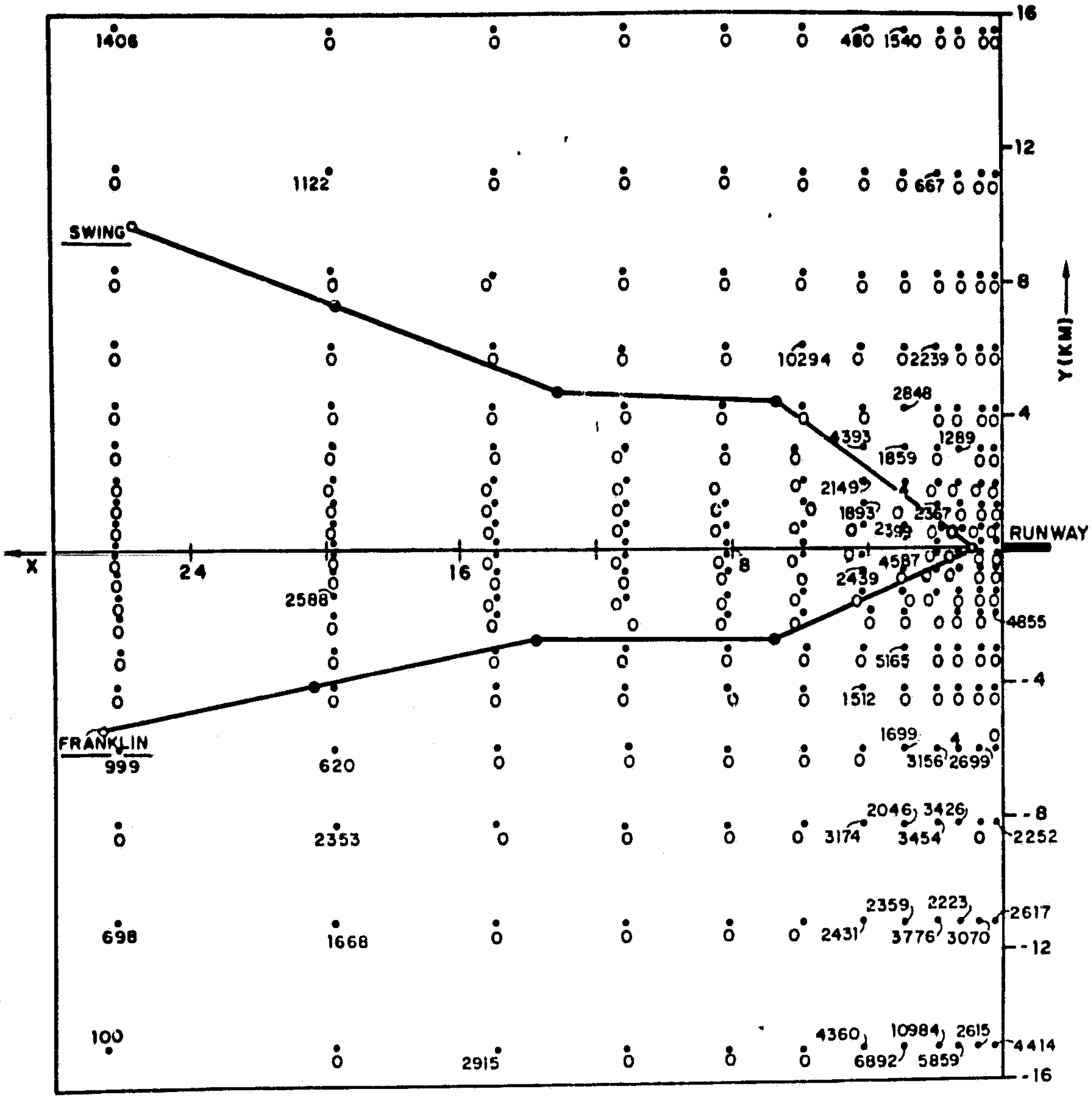


Figure 16. Four Segments per Trajectory (Landing)

Table II Line Segment Representation of Flight Path

(II) Line segment representation of flight path

(A) Balanced distribution of aircraft on each trajectory*

Number of entry points	Number of segments on each trajectory	Annoyance NII	Corner Points (x,y) Swing entry traejctory	Corner Points (x,y) Franklin entry trajectory	Corner Points (x,y) Cape Charles entry trajectory	REMARK	Figure Number
1	3	0.8395	(-25600, 9600) (-17209, 5797) (-10654, 1078) (800, 0)			Single A/C, single trajectory	13a
		0.8355		(-26666, -5333) (-18032, -3568) (-8928, -3356) (-800, 0)		Single A/C, single trajectory	13b
2	2	1.458	(-25600, 9600) (-16115, 0) (-800, 0)	(-26666,-5333) (-16115, 0) (-800, 0)		Existing approach. It violates the threshold noise & trajectories separation constraints	14a
		1.401	(-25600, 9600) (-10195, 7082) (-800, 0)	(-25555, -5333) (-11870, -4671) (-800, 0)		Optimal trajectories (with 2 segments/traj.)	14b
	3	1.395	(-25600, 9600) (-17339, 6371) (-8785, 5563) (-800, 0)	(-26666, -5333) (-17975, -3555) (-8969, -3320) (-800, 0)		Optimal trajectories (with 3 segments/traj) using centroid approximation in NII calculation.	15
	3	1.311	(-25600, 9600) (-17339, 6408) (-10031, 5752) (-800, 0)	(-26666, -5333) (-18052, -3542) (-10197, -3898) (-800, 0)		Optimal trajectories using Gaussian quadrature (3 segments/traj.)	17
	4	1.395	(-25600, 9600) (-19400, 7200) (-13181, 4861) (-6641, 4503) (-800, 0)	(-26666, -5333) (-20274, -3917) (-13708, -2718) (-6585, -2510) (-800, 0)		Optimal trajectories (with 4 segments/traj.)	16
3	3	1.262	(-25600, 9600) (-17333, 6400) (-8911, 3419) (-800, 0)	(-26666, -5333) (-18045, -3552) (-9806, -2249) (-800, 0)	(30000 -2417) (20813, -1611) (11626, -805) (2440, 0)	All three entry points used. Optimal trajectories (with 3 segments/traj.)	19a
	2	1.262	(-25600, 9600) (-16115, 0) (-800, 0)	(-26666, -5333) (-16115, 0) (-800, 0)	(30000, -2417) (16220, -1208) (2440, 0)	Existing approach	19b

*Aircraft distribution on each trajectory is:

	B707	B727
day time	2	2
night time	2	2

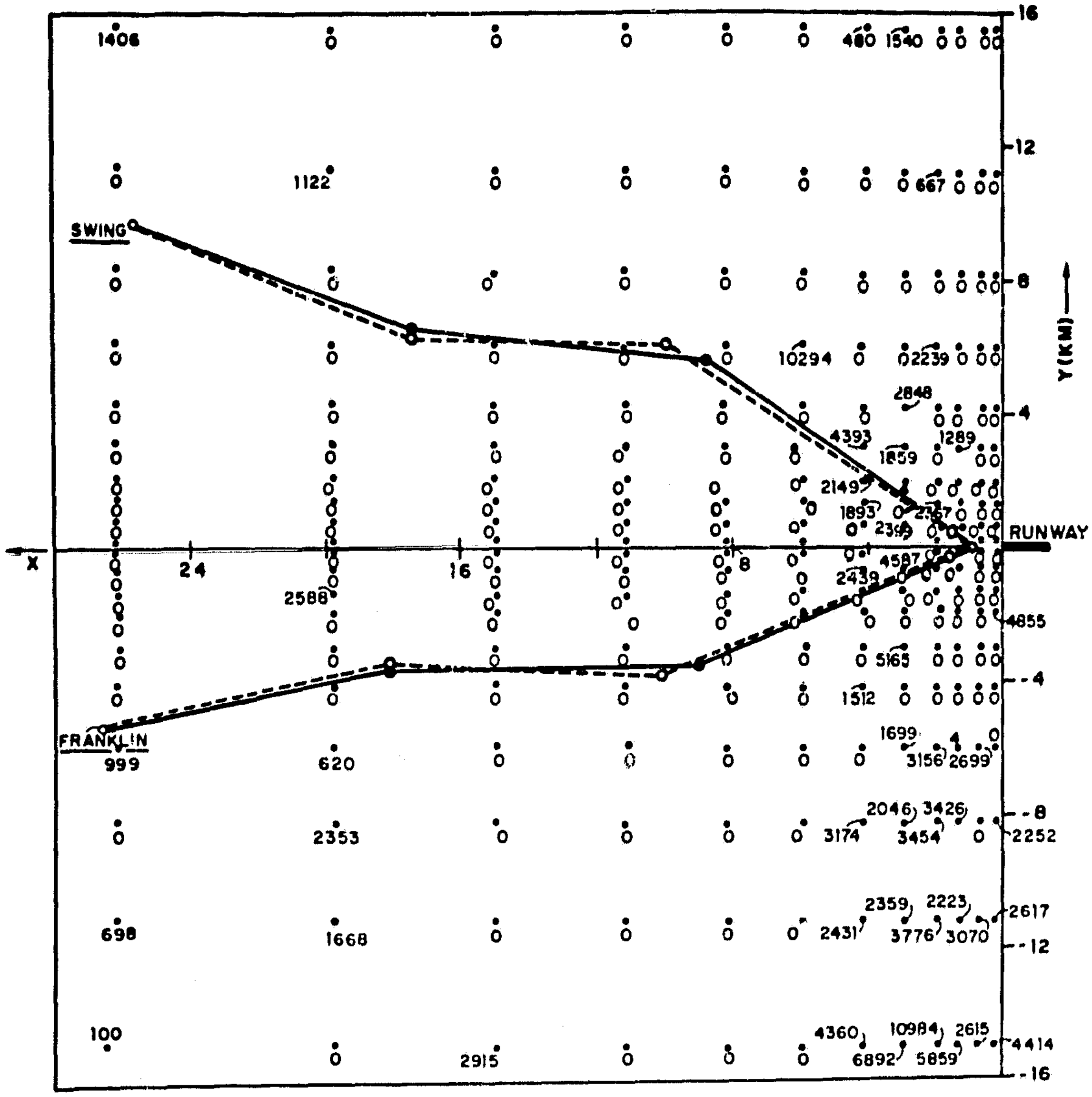


Figure 17. Comparison of Trajectories Using Gaussian Quadrature (dotted line) and Centroid Approximation (solid line)

performed as compared to the customary approximation.* Details appear in Table II.

(5) The aircraft mix on the Swing and Franklin trajectories is altered to be unbalanced (details in Table III). Figures 18a and b show that between the two cases of unbalanced aircraft distribution, the resulting change in the trajectories is slight and the change in NII is only 0.07%. This may point to the existence of optimal "corridors" which are independent of the aircraft distribution.

(6) All three entry points are used simultaneously for multiple aircraft, as shown in Figure 19. The important result here is that the optimum trajectory from Cape Charles is found to pass over the water, as should be expected. Details appear in Table II.

(7) Some preliminary work has been done on the takeoff problem. For each takeoff trajectory, the end of the runway becomes the initial point, and the final point (approximately 30 km away) may be placed anywhere. Two final points in the region northwest of the airport and two in the southwest region were chosen. Optimal paths were computed for the different pair combinations (three segments per trajectory). These are shown in Figures 20, 21, and 22 with details given in Table IV. The pair giving the lowest NII is shown in Figure 21. (NII = 1.425).

(8) To help guarantee that the optimal set of trajectories is found by the searching algorithm, a method called "selective search" has been devised. Figure 23 shows a simple version of it. Basically, a number

---

*Referred to as the centroid approximation, since the L_{dn} in a given population block is calculated at the centroid of the block and assumed constant over the entire block.

Table III Unbalanced Distribution of Aircraft

(B) Unbalanced distribution of aircraft on each trajectory*

	Swing entry trajectory					Franklin entry trajectory					Annoyance	Figure Number
	A/C distribution				Corner Points	A/C distribution				Corner Points		
	Day		Night			Day		Night				
	B707	B727	B707	B727		B707	B727	B707	B727			
CASE 1	2	2	2	2	(-25600, 9600) (-17339, 6371) (-8785, 5563) (-800, 0)	4	4	4	4	(-26666,-5333) (-17996, -3549) (-9150, -3284) (-800, 0)	1.396	18a
CASE 2	4	4	4		(-25600, 9600) (-17339, 6371) (-8785, 5563) (-800, 0)	2	2	2	2	(-26666, -5333) (-18025, -3525) (-9000, -3776) (-800, 0)	1.397	18b

*3 line segments on each trajectory

ORIGINAL PAGE IS
OF POOR QUALITY

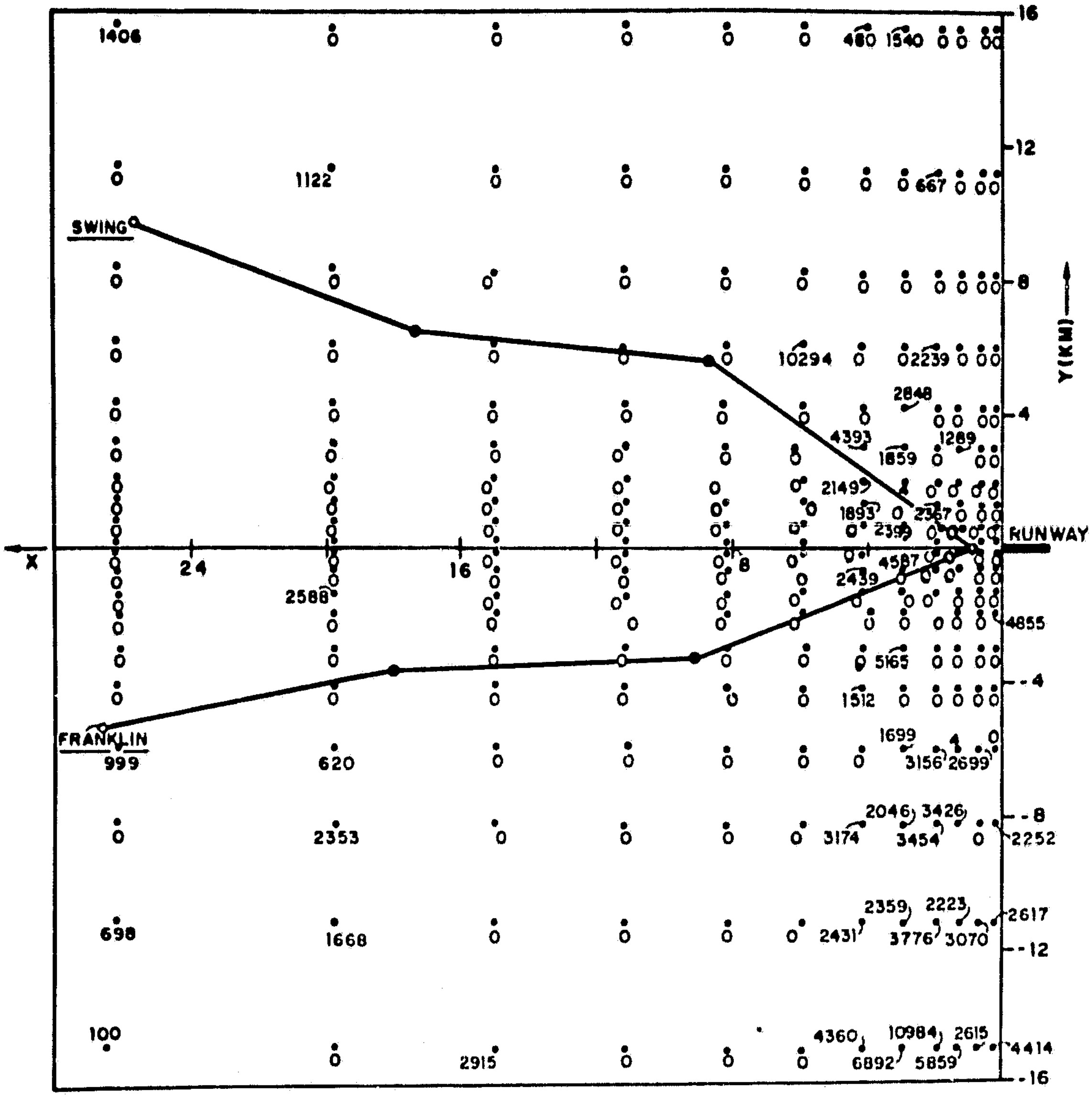


Figure 18a. Unbalanced Aircraft Distribution

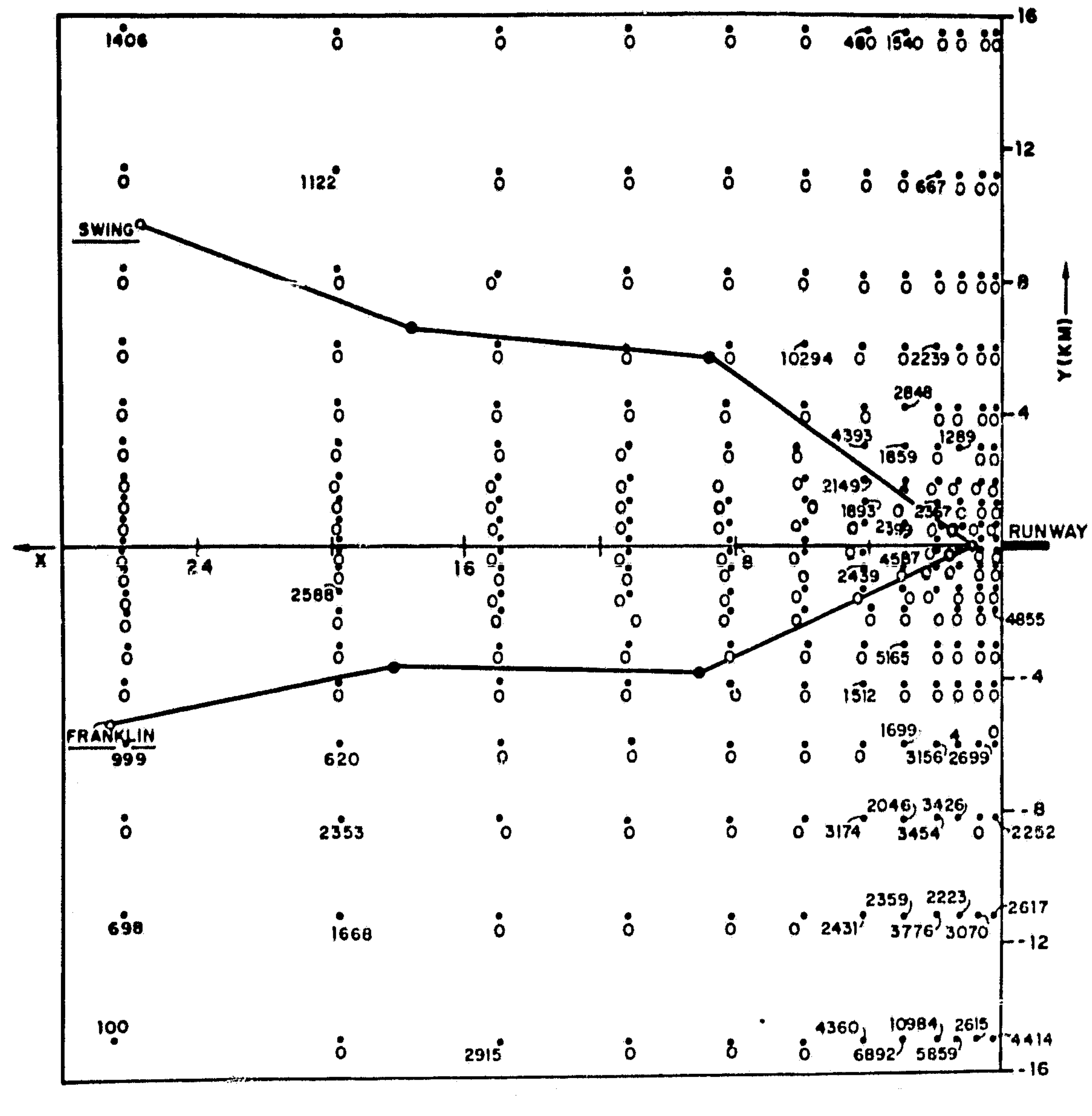


Figure 18b. Unbalanced Aircraft Distribution

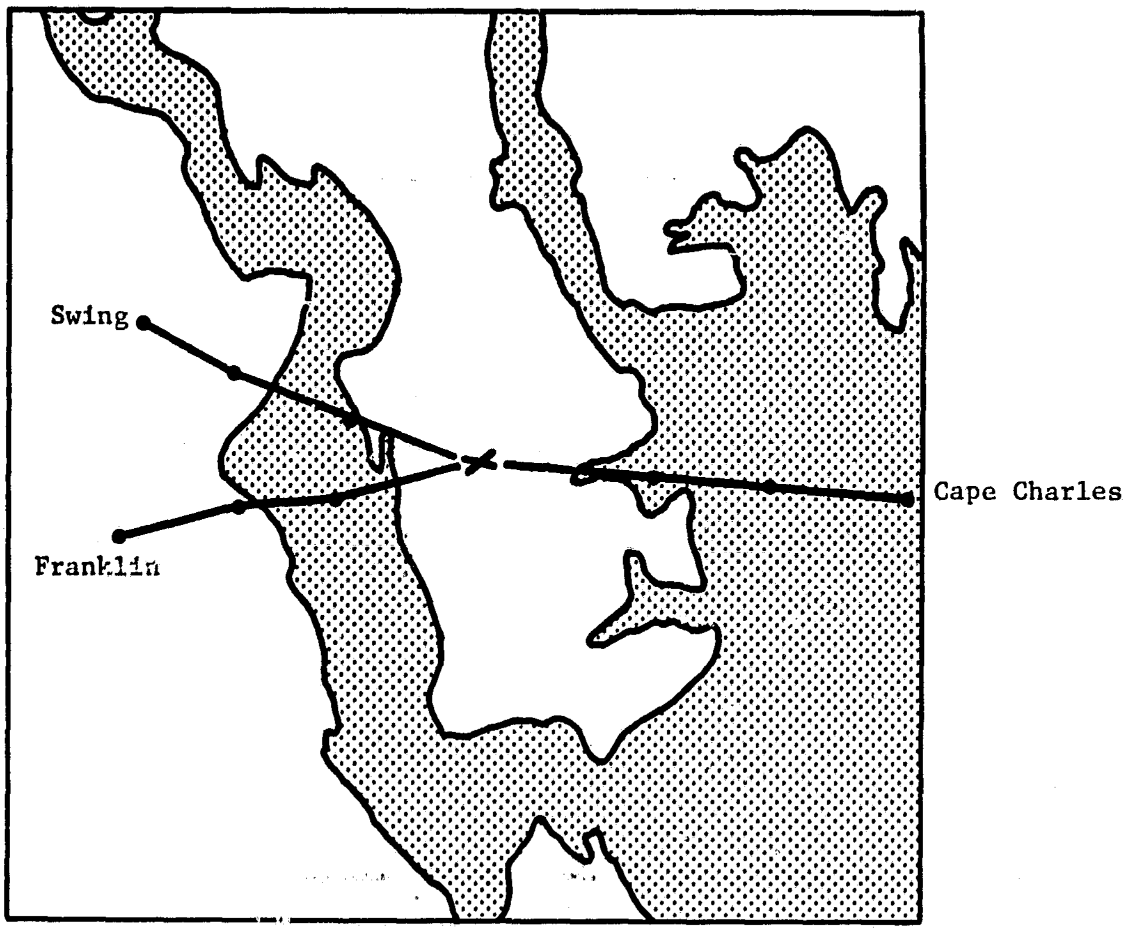


Figure 19. Three Entry Point Landing Trajectories

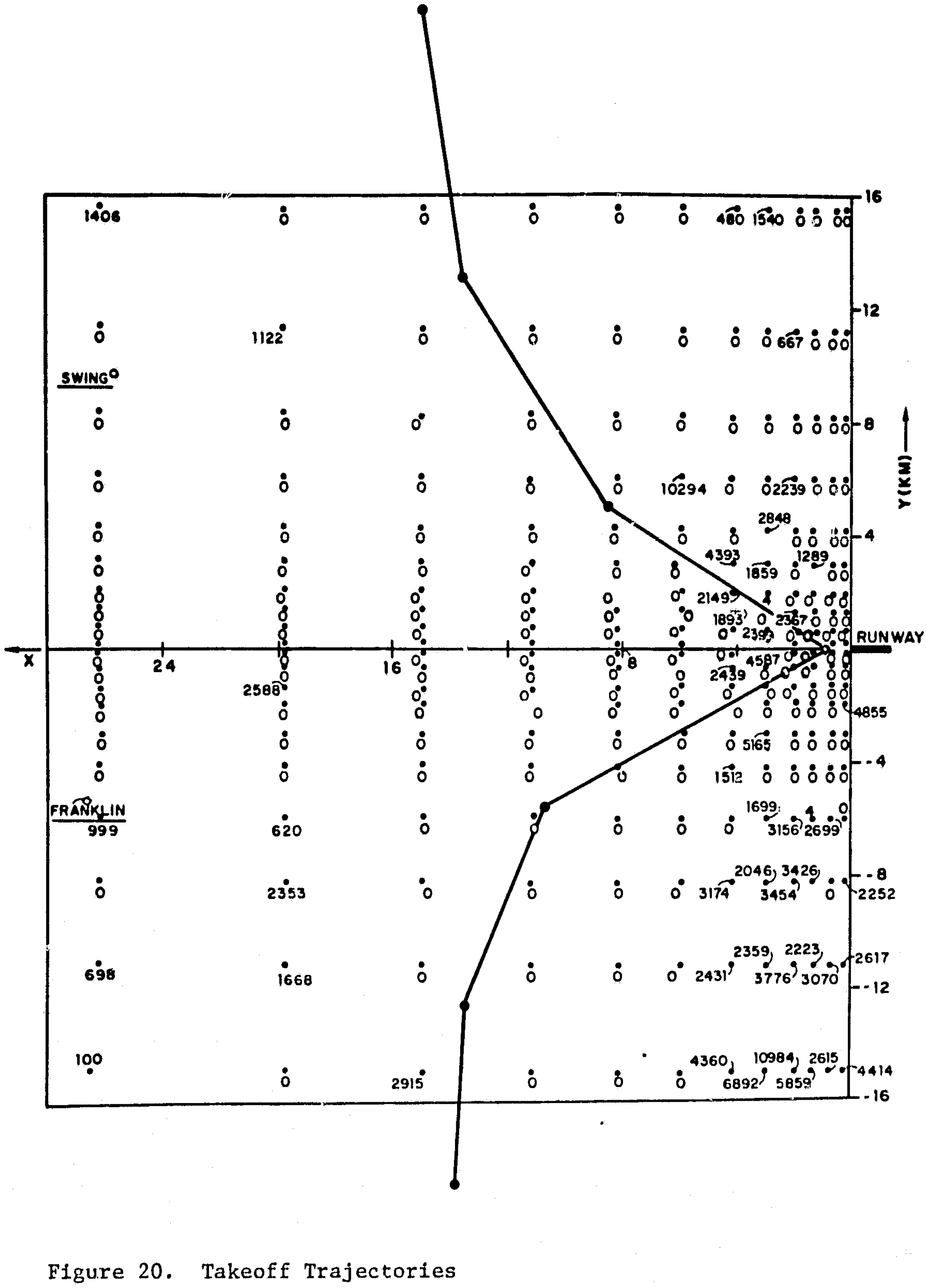


Figure 20. Takeoff Trajectories

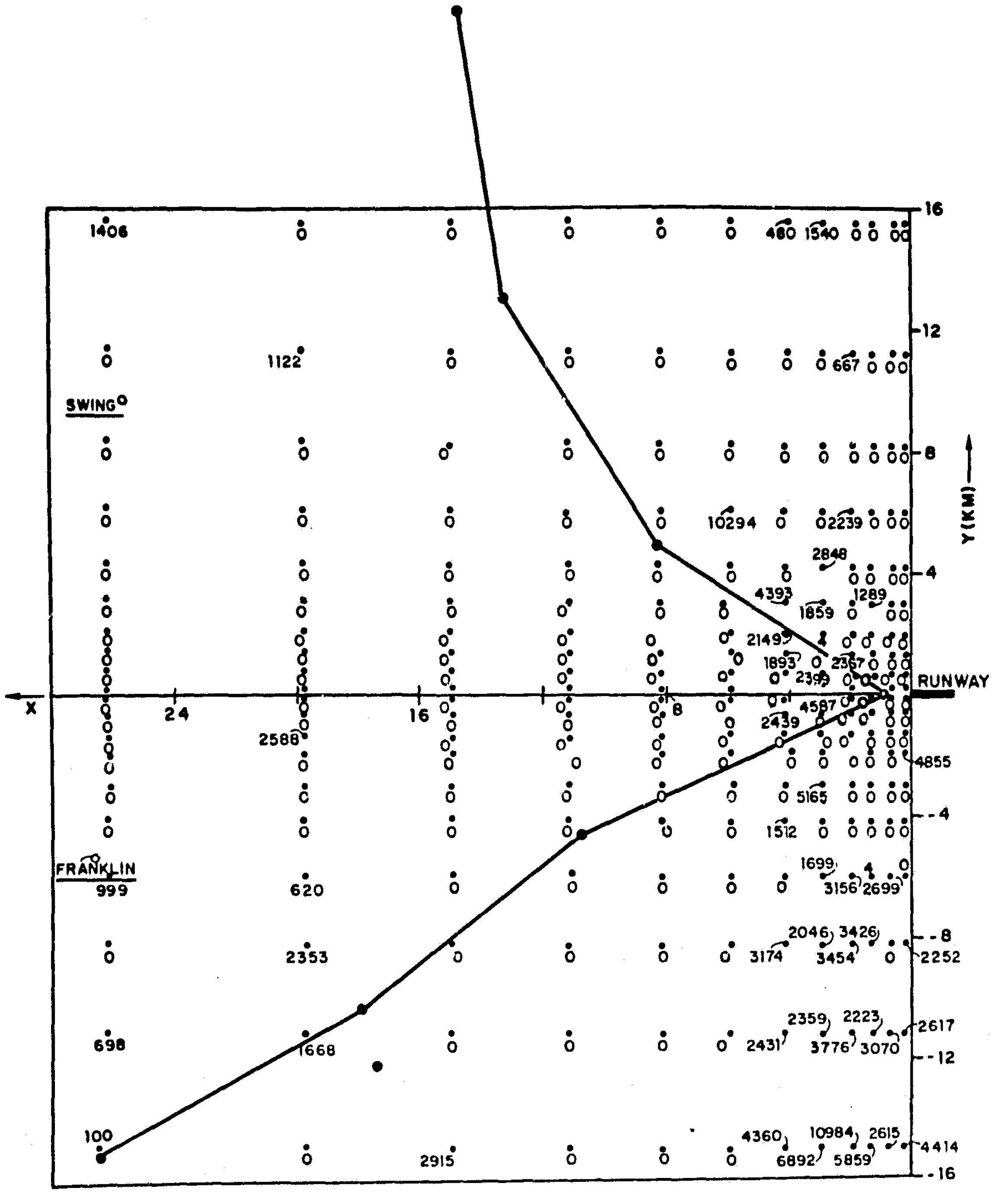


Figure 21. Takeoff Trajectories

ORIGINAL PAGE IS
OF POOR QUALITY

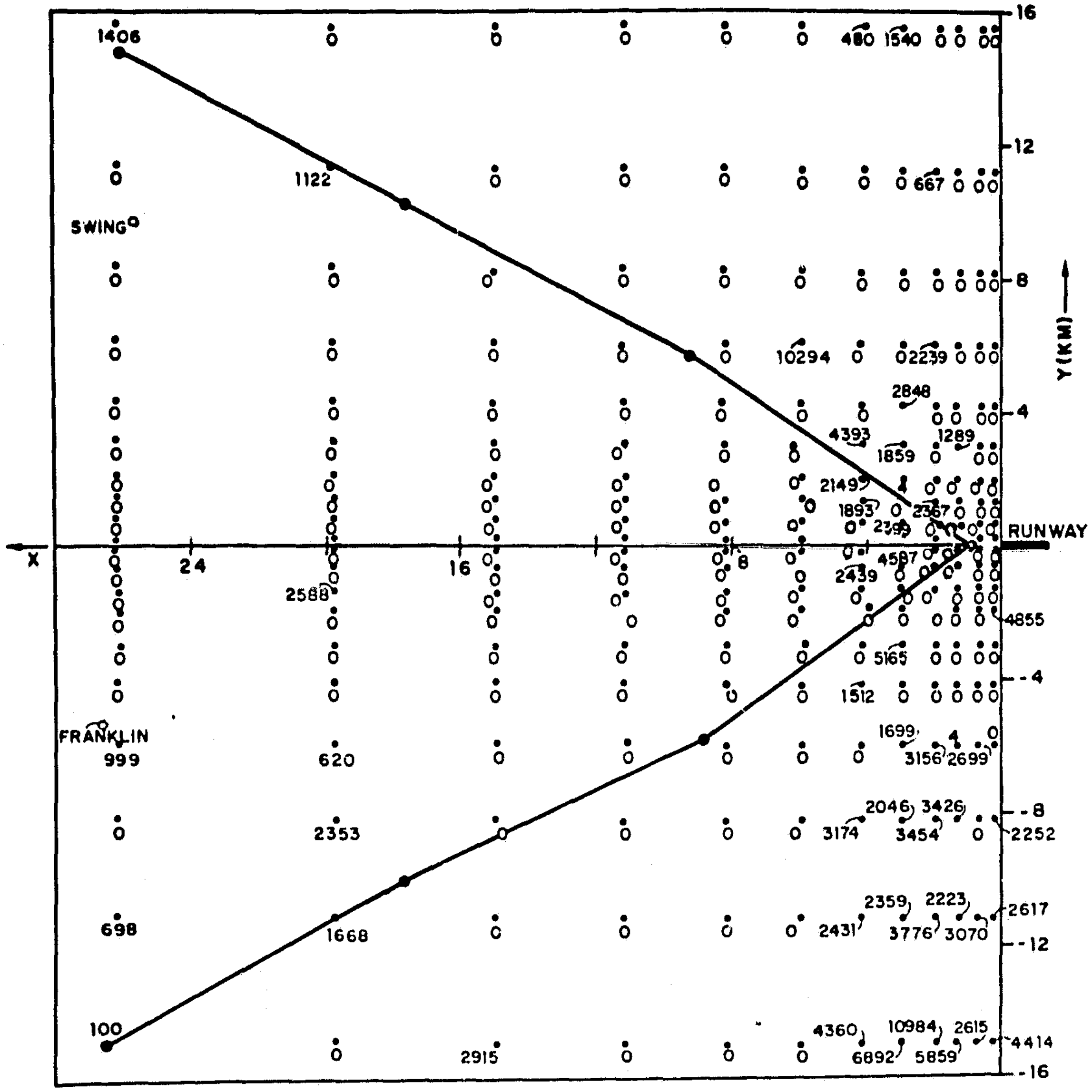


Figure 22. Takeoff Trajectories

Table IV Takeoff Trajectories

(C) Takeoff Paths*

Number of segments on each trajectory	Annoyance NII	Takeoff Trajectories				Figure Number
		Exit point 1	Exit Point 2	Exit point 3	Exit point 4	
3	1.515	(-15000, 25980) (-13200, 13200) (-8300, 5000) (-800, 0)	(-15000, -25980) (-13616, -12532) (-10894, -5341) (-800, 0)			20
3	1.425	(-15000, 25980) (-13200, 13200) (-8200, 5000) (-800, 0)		(-25980, -15000) (-17655, -10323) (-10810, -4499) (-800, 0)		21
3	1.455			(-25980, -15000) (-17565, -10030) (-8725, -5559) (-800, 0)	(-25980, 15000) (-17587, 10000) (-9204, 5363) (-800, 0)	22

*Aircraft distribution on each trajectory is

	B707	B727
day time	2	2
night time	2	2

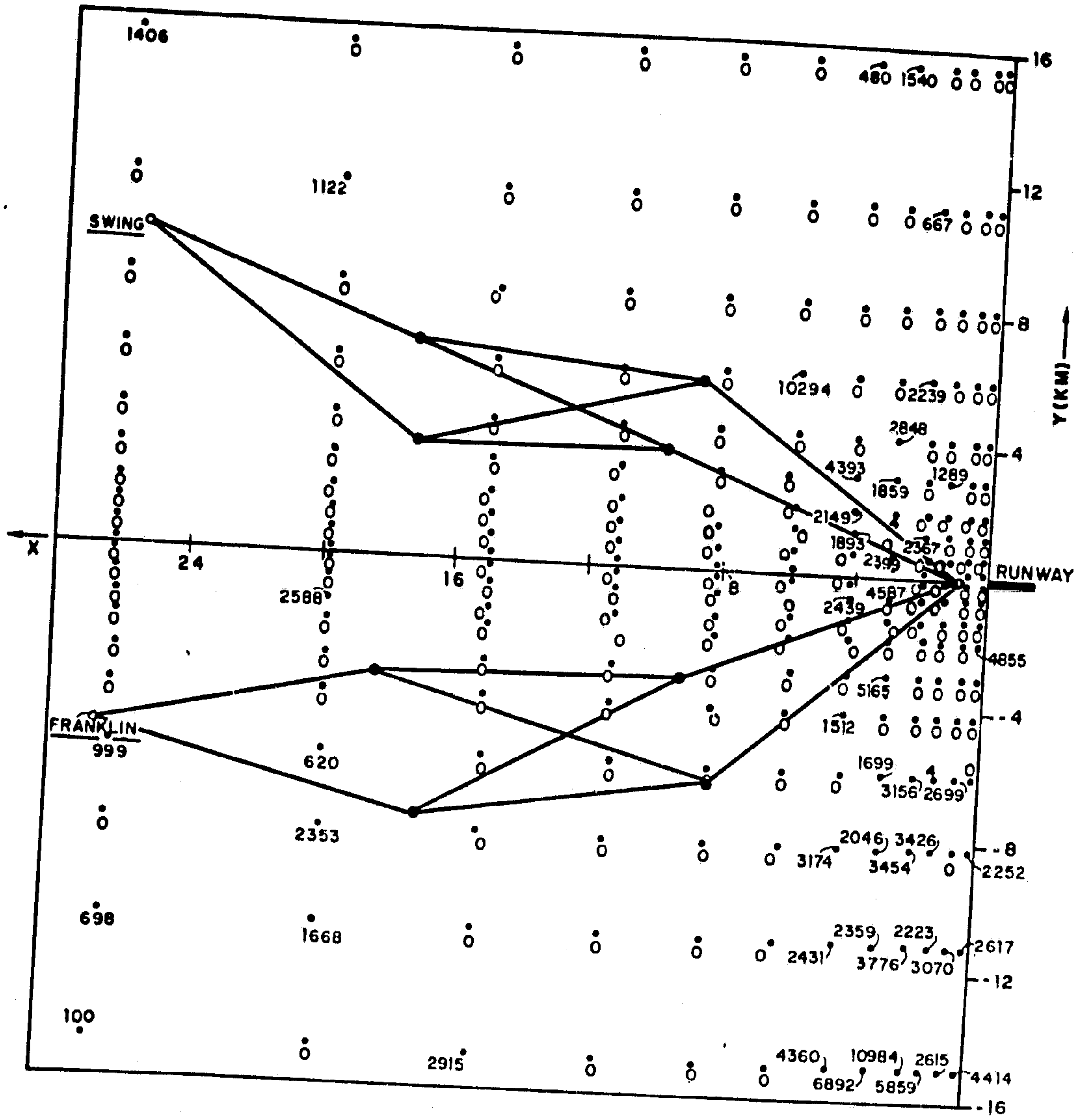


Figure 23. Selective Search Scheme

of trajectories are evaluated as candidates for the starting set input to the algorithm. The process consists of choosing several segments in different regions of the near-terminal area and evaluating the NII for various combinations of them. The set of trajectories with the lowest NII is taken as the starting set for the optimization routine. This way, it is more likely that the optimal set will be found. Such a selective search corresponds loosely with the choosing of different starting points in the example given earlier (figure 8).

CONCLUSION

A method has been formulated which optimizes aircraft paths during approach or takeoff. Multiple aircraft flying on several trajectories can be considered. Models have been developed using available data where possible for the population distribution, aircraft noise signatures, noise impact, constraints, and flight path. An algorithm which uses either the steepest descent method or the Davidon-Fletcher-Powell method for optimization has been implemented and tested. The algorithm can

1) Evaluate the noise impact of existing flight paths,
2) Evaluate the noise impact of proposed flight paths, and
3) Optimize the flight paths to minimize the noise impact, subject to constraints.

The method has been applied to the Patrick Henry International Airport area. Existing paths have been evaluated for noise impact and optimal paths were determined using either two or three of the terminal area entry points. Approximately 4.5% improvement in NII was achieved over that of the paths presently used. The population is concentrated

near the end of the runway, though, and it is felt that even more improvement in the NII could be achieved at other airports.

REFERENCES

1. Jacobson, I. D. and Cook, G., "Evaluating and Minimizing Noise Impact Due to Aircraft Flyover," University of Virginia RLES Report UVA/528166/MAE79/101, May 1979.

2. Dunn, D. G. and Peart, N. A., "Aircraft Noise Source and Contour Estimation," NASA report: CR-114649, July 1973.

3. Crowley, K. C., Jaeger, M. A., and Meldrum, D. F., "Aircraft Noise Source and Contour Computer Programs User's Guide," NASA Report: CR-114650.

4. Kolk, F. W., "A Method of Assessing the Relative Effectiveness of Various Noise Abatement Strategies," American Airline Report, June 1976.

5. Anonymous, "SITE II User's Manual," CACI, Inc., Arlington, VA, 1976.

6. Von Gierke, H. E., "Guidelines for Environmental Impact Statements with Respect to Noise," NOISE-CON 77 Proceedings, NASA Langley Research Center, 1977, pp. 61-78.

7. Luenberger, D. G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, 1973.

APPENDIX A

```
      PROGRAM MANIP (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE8,T
     1APE9)
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ARRAY(576,9),SPOSIT(3,100,3
     1),ZO(3),ZF(3),NPOSITS(3)
      COMMON /NII/ ND737(3),NN707(3),ND727(3),NN727(3)
      COMMON /DYNAMIC/ RADIUS,PWEIT1,PWEIT2,NSAMP
      COMMON /THRESH/ ALMAX,FLITMAX,TWEIT1,TWEIT2,NPLANE
      COMMON /CROSS/ XBEGIN,XFINAL,YDIS,CWEIT1,CWEIT2
      COMMON /PRINT/ SXCENTR(3,3),SYCENTR(3,3),SXT1(3,4),SYT1(3,4),SXT2(
     13,4),SYT2(3,4),SANGLE(3,3)
      DIMENSION XO(3), XF(3), YO(3), YF(3), XNOW(30,1), DELTAX(18)
C
C ..................................................................
C .                                                                .
C .    NTRAJ = NUMBER OF FLIGHT TRAJECTORIES                       .
C .    NMAP = NUMBER OF POPULATION POINTS ON MAP                   .
C .    NSEG = NUMBER OF LINE SEGMENTS ON EACH TRAJECTORY           .
C .    MAXIT = MAXIMUM ITERATION SET                               .
C .    DELH = PERTURBATION (METERS) IN X,Y DIRECTIONS AT CORNER    .
C .    POINTS FOR GRADIENT CALCULATION                             .
C .    STOPCHG = STOP CRITERION FOR SUCCESSIVE COST CHANGE         .
C .    XPORT,YPORT = AIRPORT LOCATION (METERS)                     .
C .    XO,YO,ZO,XF,YF,ZF = STARTING AND FINAL POINTS OF TRAJECTORIES .
C .    ND707,NN707,ND727,NN727 = NO. OF DAY/NIGHT B707/B727 FLIGHTS .
C .    THRESHOLD NOISE/SEPARATING CONSTRAINTS                      .
C .    PWEIT,TWEIT,CWEIT = WEIGHTING ON PENALTY OF DYNAMIC/        .
C .   THRESHOLD/CROSSOVER AND SEPARATION PENALTIES
C .    XMIN,XMAX,YMIN,YMAX = BOUNDARIES OF THE AREA WHERE THRESHOLD .
C .    CONSTRAINT IS IMPOSED                                       .
C .    XBEGIN,XFINAL = BOUNDARIES OF AREA WHERE SEPARATING         .
C .    CONSTRAINT IS IMPOSED                                       .
C .    ALMAX,RATIO = MAX. ALLOWED A-LEVEL NOISE AND PERCENTAGE OF  .
C .    FLIGHTS ALLOWED TO VIOLATE THAT LEVEL                       .
C .    YDIS = LEAST SEPARATING DISTANCE BETWEEN TRAJECTORIES ALONG Y .
C .    IF ONLY RESULT OF INITIAL CONDITION IS NEEDED,              .
C .    SET MAXIT=0                                                 .
C .                                                                .
C ..................................................................
C
      NPLANE = 0
C
C ..................................................................
C .                                                                .
C .    INPUT ALL THE INFORMATION                                   .
C .                                                                .
C ..................................................................
C
      READ (7,*) ICNTINU,NSAMP
      READ (7,*) NTRAJ,NMAP,NSEG,MAXIT
      READ (7,*) XPORT,YPORT
      READ (7,*) DELH,STOPCHG
      DO 30 I = 1,NTRAJ
        READ (7,*) XO(I),YO(I),ZO(I)
        READ (7,*) XF(I),YF(I),ZF(I)
        READ (7,*) ND707(I),NN707(I),ND727(I),NN727(I)
C
C ..................................................................
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C .                                                             .
      C .   CALCULATE INITIAL CORNER POINTS                           .
      C .                                                             .
      C ...............................................................
      C
            NPLANE = NPLANE+ND707(I)+NN707(I)+ND727(I)+NN727(I)
            XM(I,1) = XO(I)
            YM(I,1) = YO(I)
            XM(I,NSEG+1) = XF(I)
            YM(I,NSEG+1) = YF(I)
            XM(I,NSEG+2) = (XPORT+XF(I))/2.
            YM(I,NSEG+2) = (YPORT+YF(I))/2.
            IF (ICNTINU.EQ.1) GO TO 20
            DO 10 J = 2,NSEG
              XM(I,J) = XO(I)+(FLOAT(J-1)/FLOAT(NSEG))*(XF(I)-XO(I))
              YM(I,J) = YO(I)+(FLOAT(J-1)/FLOAT(NSEG))*(YF(I)-YO(I))
         10   CONTINUE
            GO TO 30
         20 READ (7,*) (XM(I,J),YM(I,J),J=2,NSEG)
         30 CONTINUE
          DO 40 I = 1,NMAP
            DO 40 J = 4,6
            ARRAY(I,J) = 0.
         40 CONTINUE
          READ (7,*) ((ARRAY(I,J),J=1,9),I=1,NMAP)
          READ (7,*) PWEIT1,PWEIT2,RADIUS
          READ (7,*) XMIN,XMAX,YMIN,YMAX
          READ (7,*) ALMAX,RATIO,TWEIT1,TWEIT2
          READ (7,*) XBEGIN,XFINAL,YDIS,CWEIT1,CWEIT2
          DO 50 I = 1,NMAP
            ARRAY(I,6) = 0.
            IF (XMIN.LE.ARRAY(I,1).AND.ARRAY(I,1).LE.XMAX.AND.YMIN.LE.ARRAY(
           1 I,2).AND.ARRAY(I,2).LE.YMAX.AND.ARRAY(I,3).NE.0.) ARRAY(I,6) = 1
           2 .
         50 CONTINUE
          FLITMAX = NPLANE*RATIO
      C
      C ...............................................................
      C .                                                             .
      C .   PRINT INFORMATION INPUT                                   .
      C .                                                             .
      C ...............................................................
      C
          WRITE (6,9010) MAXIT,NTRAJ,NSEG,NMAP,XPORT,YPORT,DELH,STOPCHG
          WRITE (6,9020) XMIN,XMAX,YMIN,YMAX,ALMAX,FLITMAX
          WRITE (6,9030) XBEGIN,XFINAL,YDIS
          DO 70 I = 1,NTRAJ
            WRITE (6,9040) I,XO(I),YO(I),ZO(I),XF(I),YF(I),ZF(I)
            NSEG1 = NSEG+1
            DO 60 J = 1,NSEG1
              WRITE (6,9050) J,XM(I,J),YM(I,J)
         60   CONTINUE
            WRITE (6,9060) ND707(I),ND727(I),NN707(I),NN727(I)
         70 CONTINUE
          N = 2*(NSEG-1)*NTRAJ
          DO 80 I = 1,NTRAJ
            NSEG1 = NSEG-1
```

ORIGINAL PAGE IS OF POOR QUALITY


```
              DO 80 J = 1,NSEG1
              DO 80 K = 1,2
              L = (I-1)*2*(NSEG-1)+(J-1)*2+K
              IF (K.EQ.1) XNOW(L,1) = XM(I,J+1)
              IF (K.EQ.2) XNOW(L,1) = YM(I,J+1)
         80   CONTINUE
            DO 90 I = 1,N
              DELTAX(I) = DELH
         90   CONTINUE
      C
      C ..............................................................
      C .                                                            .
      C .   START OPTIMIZATION                                       .
      C .                                                            .
      C ..............................................................
      C
            CALL QNEWTON (MAXIT,STOPCHG,N,XNOW,DELTAX)
            STOP
      C
       9010 FORMAT (1H1,4X,23HMAXIMUM ITERATION SET: ,I2,/,5X,24HNUMBER OF TRA
           1JECTORIES: ,I1,/,5X,39HNUMBER OF SEGMENTS ON EACH TRAJECTORY: ,I1,
           2/,5X,40HNUMBER OF POPULATION POINTS ON THE MAP: ,I3,/,5X,39HAIRPOR
           3T RUNWAY LOCATION X,Y IN METERS: ,F8.1,10X,F8.1,/,5X,37HPERTURB TR
           4AJECTORIES IN Y DIRECTIONS ,F10.5,11H METERS FOR,1X,21HCALCULATING
           5 GRADIENTS,/,5X,28HEXIT CRITERION FOR QNEWTON: ,F7.5,/)
       9020 FORMAT (5X,37HFOR BLOCKS WITH X-COORDINATE BETWEEN ,F8.1,5H AND ,F
           18.1,21H YCOORDINATE BETWEEN ,F8.1,5H AND ,F6.1,9H AND WITH,/,7X,55
           2HPOPULATION NOT EQUAL TO ZERO SHOULD RECEIVE NOISE OVER ,F6.1,26H
           3A-LEVEL DB NOT MORE THAN ,F4.1,12H TIMES A DAY,/)
       9030 FORMAT (5X,40HWITHIN THE AREA OF X-COORDINATES BETWEEN,F6.1,5H AND
           1 ,F6.1,41H, THE SEPARATING DISTANCE BETWEEN TRAJECT,5HORIES,/,5X,1
           28HSHOULD BE AT LEAST,F6.1,7H METERS,//,5X,31HINFORMATION OF EACH T
           3RAJECTORY:,//)
       9040 FORMAT (10X,15HFLIGHTPATH NO: ,I1,/,12X,13HINITIAL X,Y,Z,1X,23HCOO
           1RDINATES IN METERS: ,3(F8.1,3X),/,12X,11HFINAL X,Y,Z,1X,23HCOORDIN
           2ATES IN METERS: ,3(F8.1,3X),/,12X,14HINITIAL CORNER,18H POINT POSI
           3TIONS: ,/,14X,10HCORNER NO.,6X,1HX,9X,1HY,/)
       9050 FORMAT (18X,I1,6X,F8.1,3X,F8.1)
       9060 FORMAT (10X,14HAIRCRAFT TYPE:,5X,4HB707,5X,4HB727,/,16X,7HDAYTIME,
           17X,I2,7X,I2,/,14X,9HNIGHTTIME,7X,I2,7X,I2,//)
            END

42000B CM STORAGE USED          1.408 SECONDS
```

```
      SUBROUTINE COST (IGPAD,TOTAL,ANII,PNALTY,CLOSE,THRESH)
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ARRAY(576,9),SPOSIT(3,100,3
     1),ZO(3),ZF(3),NPOSITS(3)
      COMMON /NII/ ND707(3),NN707(3),ND727(3),NN727(3)
      COMMON /DYNAMIC/ RADIUS,PWEIT1,PWEIT2,NSAMP
      COMMON /THRESH/ ALMAX,FLITMAX,TWEIT1,TWEIT2,NPLANE
      COMMON /PRINT/ SXCENTR(3,3),SYCENTR(3,3),SXT1(3,4),SYT1(3,4),SXT2(
     13,4),SYT2(3,4),SANGLE(3,3)
      DIMENSION DIS(3), ANGLE(3,3), LOCAL(576,8), POSIT(3,100,3), XCENTR
     1(3,3), YCENTR(3,3), XT1(3,4), YT1(3,4), XT2(3,4), YT2(3,4), NPOSIT
     2(3)
      REAL LOCAL
      THRESH = 0.
      CLOSE = THRESH
      PNALTY = CLOSE
      ANII = PNALTY
      PI = ATAN(1.)*4.
C
C ................................................................
C .                                                              .
C .   FIRST CALCULATE PARAMETERS AT CORNER POINTS                .
C .                                                              .
C ................................................................
C
      DO 50 I = 1,NTRAJ
        DO 10 J = 2,NSEG
          YT2(I,J) = 0.
          YT1(I,J) = YT2(I,J)
          XT2(I,J) = YT1(I,J)
          XT1(I,J) = XT2(I,J)
          ANGLE(I,J-1) = XT1(I,J)
          YCENTR(I,J-1) = ANGLE(I,J-1)
          XCENTR(I,J-1) = YCENTR(I,J-1)
   10     CONTINUE
        DIS(1) = 0.
        XT1(I,1) = XM(I,1)
        XT2(I,1) = XM(I,1)
        YT1(I,1) = YM(I,1)
        YT2(I,1) = YM(I,1)
        DO 40 J = 2,NSEG
          A = SQRT((XM(I,J-1)-XM(I,J))**2+(YM(I,J-1)-YM(I,J))**2)
          B = SQRT((XM(I,J)-XM(I,J+1))**2+(YM(I,J)-YM(I,J+1))**2)
          C = SQRT((XM(I,J-1)-XM(I,J+1))**2+(YM(I,J-1)-YM(I,J+1))**2)
C
C ................................................................
C .                                                              .
C .   DO THE TWO SEGMENTS FORM A STRAIGHT LINE+                  .
C .                                                              .
C ................................................................
C
          IF (ABS(A+B-C).GT.1.) GO TO 20
          SEG = SQRT((XT2(I,J-1)-XM(I,J))**2+(YT2(I,J-1)-YM(I,J))**2)
          XT1(I,J) = XM(I,J)
          XT2(I,J) = XM(I,J)
          YT1(I,J) = YM(I,J)
          YT2(I,J) = YM(I,J)
          YCENTR(I,J-1) = 0.
```

```
      XCENTR(I,J-1) = YCENTR(I,J-1)
      ALONG1 = 0.
      ANGLE(I,J-1) = PI
      GO TO 30
20    A = SQRT((XM(I,J-1)-XM(I,J))**2+(YM(I,J-1)-YM(I,J))**2)
      B = SQRT((XM(I,J)-XM(I,J+1))**2+(YM(I,J)-YM(I,J+1))**2)
      C = SQRT((XM(I,J-1)-XM(I,J+1))**2+(YM(I,J-1)-YM(I,J+1))**2)
      BETA = AACOS(A,B,C)
      ALONG2 = RADIUS/TAN(BETA/2.)
      PNALTY = PNALTY+PWEIT1*((AMAX1(0.,(ALONG2-A)))**2+(AMAX1(0.,(A
     1LONG2-B)))**2)
      IF (J.NE.2) PNALTY = PNALTY+PWEIT1*(AMAX1(0.,(ALONG1+ALONG2-A)
     1)**2)
C
C ..................................................................
C .                                                                .
C .   CALCULATE TANGENTIAL POINTS, CENTER OF RADIUS R AT J-TH CORNER .
C .                                                                .
C ..................................................................
C
      XT1(I,J) = XM(I,J)+ALONG2*(XM(I,J-1)-XM(I,J))/A
      YT1(I,J) = YM(I,J)+ALONG2*(YM(I,J-1)-YM(I,J))/A
      XT2(I,J) = XM(I,J)+ALONG2*(XM(I,J+1)-XM(I,J))/B
      YT2(I,J) = YM(I,J)+ALONG2*(YM(I,J+1)-YM(I,J))/B
      SEG = SQRT((XT2(I,J-1)-XT1(I,J))**2+(YT2(I,J-1)-YT1(I,J))**2)
      DISM2 = ALONG2*COS(BETA/2.)
      DISC2 = RADIUS/SIN(BETA/2.)
      X = (XT1(I,J)+XT2(I,J))/2.
      Y = (YT1(I,J)+YT2(I,J))/2.
      XCENTR(I,J-1) = XM(I,J)+DISC2*(X-XM(I,J))/DISM2
      YCENTR(I,J-1) = YM(I,J)+DISC2*(Y-YM(I,J))/DISM2
      ALONG1 = ALONG2
      D = SQRT((XT1(I,J)-XT2(I,J))**2+(YT1(I,J)-YT2(I,J))**2)
      ANGLE(I,J-1) = AACOS(RADIUS,RADIUS,D)
      SEG = SEG+ANGLE(I,J-1)*RADIUS
30    DIS(I) = DIS(I)+SEG
40    CONTINUE
      A = SQRT((XM(I,NSEG)-XM(I,NSEG+1))**2+(YM(I,NSEG)-YM(I,NSEG+1))*
     1*2)
      B = SQRT((XM(I,NSEG+1)-XM(I,NSEG+2))**2+(YM(I,NSEG+1)-YM(I,NSEG+
     12))**2)
      C = SQRT((XM(I,NSEG)-XM(I,NSEG+2))**2+(YM(I,NSEG)-YM(I,NSEG+2))*
     1*2)
      ANG = AACOS(A,B,C)
      PNALTY = PNALTY+PWEIT1*(AMAX1(0.,(400.+ALONG1-A))**2)+PWEIT2*(AM
     1AX1(0.,(2.380579899-ANG))**2)
      DIS(I) = DIS(I)+SQRT((XT2(I,NSEG)-XM(I,NSEG+1))**2+(YT2(I,NSEG)-
     1YM(I,NSEG+1))**2)
50    CONTINUE
C
C ..................................................................
C .                                                                .
C .   RETURN IF DYNAMIC CONSTRAINT NOT SATISFIED                   .
C .                                                                .
C ..................................................................
C
      IF (PNALTY.EQ.0.) GO TO 60
```

```
      TOTAL = PNALTY
      RETURN
C
C ...................................................................
C .                                                                 .
C .   TAKE SAMPLES ALONG THE TRAJECTORY                             .
C .                                                                 .
C ...................................................................
C
   60 DO 120 I = 1,NTRAJ
        DELZPM = (ZO(I)-ZF(I))/DIS(I)
        K = 1
        POSIT(I,1,1) = XM(I,1)
        POSIT(I,1,2) = YM(I,1)
        POSIT(I,1,3) = ZO(I)
        DO 100 J = 2,NSEG
          DELX = (XT2(I,J-1)-XT1(I,J))/NSAMP
          DELY = (YT2(I,J-1)-YT1(I,J))/NSAMP
          DO 70 L = 1,NSAMP
            POSIT(I,K+1,1) = POSIT(I,K,1)-DELX
            POSIT(I,K+1,2) = POSIT(I,K,2)-DELY
            POSIT(I,K+1,3) = POSIT(I,K,3)-DELZPM*SQRT(DELX**2+DELY**2)
            K = K+1
   70       CONTINUE
          IF (ANGLE(I,J-1).EQ.PI) GO TO 100
C
C ...................................................................
C .                                                                 .
C .   SAMPLE THE CORRECT ARC ON THE J-1 TH CIRCLE                   .
C .                                                                 .
C ...................................................................
C
          ALFA1 = ATAN2((YT1(I,J)-YCENTR(I,J-1)),(XT1(I,J)-XCENTR(I,J-1)
     1    ))
          IPLUS = 1
          AA = ALFA1+ANGLE(I,J-1)/4.
          X22 = XCENTR(I,J-1)+RADIUS*COS(AA)
          Y22 = YCENTR(I,J-1)+RADIUS*SIN(AA)
          ADIS = SQRT((X22-XM(I,J))**2+(Y22-YM(I,J))**2)
          BDIS = SQRT((XT1(I,J)-XM(I,J))**2+(YT1(I,J)-YM(I,J))**2)
          IF (BDIS.GT.ADIS) IPLUS = -1
          II = 0
          A = ANGLE(I,J-1)
   80     A = A-.1745329252
          IF (A.LE.0.) GO TO 90
          II = II+1
          AA = ALFA1+IPLUS*II*.1745329252
          POSIT(I,K+1,1) = XCENTR(I,J-1)+RADIUS*COS(-AA)
          POSIT(I,K+1,2) = YCENTR(I,J-1)-RADIUS*SIN(-AA)
          D = RADIUS*.1745329252
          POSIT(I,K+1,3) = POSIT(I,K,3)-DELZPM*D
          K = K+1
          GO TO 80
   90     POSIT(I,K+1,1) = XT2(I,J)
          POSIT(I,K+1,2) = YT2(I,J)
          D = RADIUS*(ANGLE(I,J-1)-FLOAT(II)*0.1745329252)
          POSIT(I,K+1,3) = POSIT(I,K,3)-DELZPM*D
```

```
            K = K+1
  100       CONTINUE
          DELX = (XT2(I,NSEG)-XM(I,NSEG+1))/NSAMP
          DELY = (YT2(I,NSEG)-YM(I,NSEG+1))/NSAMP
          DO 110 L = 1,NSAMP
            POSIT(I,K+1,1) = POSIT(I,K,1)-DELX
            POSIT(I,K+1,2) = POSIT(I,K,2)-DELY
            POSIT(I,K+1,3) = POSIT(I,K,3)-DELZPM*SQRT(DELX**2+DELY**2)
            K = K+1
  110       CONTINUE
          NPOSIT(I) = K
  120     CONTINUE
        IF (NTRAJ.EQ.1) GO TO 270
        DO 130 I = 1,NMAP
          DO 130 J = 4,5
          DO 130 K = 3,8
          LOCAL(I,K) = 0.
          ARRAY(I,J) = LOCAL(I,K)
  130     CONTINUE
        DO 200 I = 1,NTRAJ
          DO 140 K = 1,NMAP
            LOCAL(K,2) = 0.
            LOCAL(K,1) = LOCAL(K,2)
  140       CONTINUE
C
C ***************************************************************
C *                                                             *
C *   CALCULATE ANNOYANCE LEVEL                                 *
C *   ANY BLOCK WITH ZERO POPULATION IS BYPASSED                *
C *   IN "LOCAL":   COL1 = HIGHEST 707 NOISE                    *
C *                 COL2 = HIGHEST 727 NOISE                    *
C *                                                             *
C ***************************************************************
C
          NPOSITI = NPOSIT(I)
          DO 160 J = 1,NPOSITI
            DO 160 K = 1,NMAP
            IF (ARRAY(K,3).EQ.0.) GO TO 160
            RANGE = SQRT((POSIT(I,J,1)-ARRAY(K,1))**2+(POSIT(I,J,2)-ARRAY(
     1      K,2))**2+POSIT(I,J,3)**2)
            AL707 = 129.-25.*ALOG10(3.281*RANGE/200.)
            AL727 = 114.-22.5*ALOG10(3.281*RANGE/500.)
            IF (AL707.LE.LOCAL(K,1)) GO TO 150
            LOCAL(K,1) = AL707
  150       IF (AL727.LE.LOCAL(K,2)) GO TO 160
            LOCAL(K,2) = AL727
  160       CONTINUE
          DO 190 K = 1,NMAP
C
C ***************************************************************
C *                                                             *
C *   IN "LOCAL":  COL 3-5 = NO. OF OCCURANCES HIGHER THAN ALMAX  *
C *   DUE TO TRAJ. NUMBERS 1-3 RESPECTIVELY                     *
C *              COL 6-8 = TOTAL VIOLATING NOISE - ALMAX FOR TRAJ. *
C *   1-3 RESPECTIVELY                                          *
C *                                                             *
C ***************************************************************
```

```
C
            IF (ARRAY(K,3).EQ.0.) GO TO 190
            IF (ARRAY(K,6).NE.1) GO TO 190
            IF (LOCAL(K,1).LE.ALMAX) GO TO 170
            LOCAL(K,I+2) = ND707(I)+NN707(I)
            LOCAL(K,I+5) = (LOCAL(K,1)-ALMAX)*(ND707(I)+NN707(I))
  170       IF (LOCAL(K,2).LE.ALMAX) GO TO 180
            LOCAL(K,I+2) = LOCAL(K,I+2)+ND727(I)+NN727(I)
            LOCAL(K,I+5) = LOCAL(K,I+5)+(LOCAL(K,2)-ALMAX)*(ND727(I)+NN727
     1      (I))
  180       IF (LOCAL(K,I+2).LE.FLITMAX) GO TO 190
            THRESH = THRESH+TWEIT1*(AMAX1(0.,(LOCAL(K,I+2)-FLITMAX))**2)+T
     1      WEIT2*(AMAX1(0.,LOCAL(K,I+5))**2)
  190       CONTINUE
  200     CONTINUE
        DO 210 I = 1,NTRAJ
          PEOPLE = 0.
          DO 210 K = 1,NMAP
          IF (ARRAY(K,3).EQ.0.) GO TO 210
          ARRAY(K,4) = ARRAY(K,4)+ND707(I)*10.**(LOCAL(K,1)/10.)+NN707(I)*
     1    10.**(1.+LOCAL(K,1)/10.)+ND727(I)*10.**(LOCAL(K,2)/10.)+NN727(I)
     2    *10.**(1.+LOCAL(K,2)/10.)
          PEOPLE = PEOPLE+ARRAY(K,3)
  210     CONTINUE
C
C ****************************************************************
C *                                                              *
C *    CHECK THRESHOLD NOISE CONSTRAINTS                         *
C *                                                              *
C ****************************************************************
C
        DO 220 K = 1,NMAP
          IF (ARRAY(K,3).EQ.0.) GO TO 220
          AVN = 10.*ALOG10(ARRAY(K,4)/NPLANE)
          ARRAY(K,4) = AVN
          IF (AVN.LT.20..OR.ARRAY(K,3).EQ.0.) GO TO 220
          ARRAY(K,5) = 3.36E-6*10.**(.103*AVN)/(.2*10.**(.03*AVN)+1.43E-4*
     1    10.**(.06*AVN))
          ARRAY(K,5) = ARRAY(K,3)*ARRAY(K,5)/PEOPLE
          ANII = ANII+ARRAY(K,5)
  220     CONTINUE
C
C ****************************************************************
C *                                                              *
C *    CHECK FOR THRESHOLD NOISE VIOLATION AT EACH BLOCK DUE     *
C *    TO COMBINATION EFFECT OF MULTIPLE FLYOVER                 *
C *                                                              *
C ****************************************************************
C
        DO 260 I = 1,NMAP
          IF (ARRAY(I,6).NE.1.) GO TO 260
          VIOLA = 0.
          DO 230 J = 1,NTRAJ
            VIOLA = VIOLA+LOCAL(I,J+2)
  230       CONTINUE
          IF (VIOLA.LE.FLITMAX) GO TO 260
          DO 240 J = 1,NTRAJ
```

```
          IF (LOCAL(I,J+2).GT.FLITMAX) GO TO 260
  240     CONTINUE
        SUM = 0.
        DO 250 J = 1,NTRAJ
          SUM = SUM+LOCAL(I,J+5)
  250     CONTINUE
        THRESH = THRESH+TWEIT1*(AMAX1(0.,(VIOLA-FLITMAX))**2)+TWEIT2*(AM
     1  AX1(0.,SUM)**2)
  260   CONTINUE
C
C ****************************************************************
C *                                                              *
C *    CHECK SEPARATING CONSTRAINTS                              *
C *                                                              *
C ****************************************************************
C
      IF (NTRAJ.NE.1) CALL CROSOVR (NPOSIT,POSIT,CLOSE)
      GO TO 320
  270 DO 280 I = 1,NMAP
        DO 280 J = 1,2
        LOCAL(I,J) = 0.
  280   CONTINUE
      DO 290 I = 1,K
        DO 290 J = 1,NMAP
        IF (ARRAY(J,3).EQ.0.) GO TO 290
        RANGE = SQRT((POSIT(1,I,1)-ARRAY(J,1))**2+(POSIT(1,I,2)-ARRAY(J,
     1  2))**2+POSIT(1,I,3)**2)
        AL707 = 129.-25.*ALOG10(3.281*RANGE/200.)
        AL727 = 115.-22.5*ALOG10(3.281*RANGE/500.)
        LOCAL(J,1) = AMAX1(AL707,LOCAL(J,1))
        LOCAL(J,2) = AMAX1(AL727,LOCAL(J,2))
  290   CONTINUE
      PEOPLE = 0.
      DO 300 I = 1,NMAP
        IF (ARRAY(I,3).EQ.0.) GO TO 300
        ARRAY(I,4) = ARRAY(I,4)+ND707(1)*10.**(LOCAL(I,1)/10.)+NN707(1)*
     1  10.**(1.+LOCAL(I,1)/10.)+ND727(1)*10.**(LOCAL(I,2)/10.)+NN727(1)
     2  *10.**(1.+LOCAL(I,2)/10.)
        PEOPLE = PEOPLE+ARRAY(I,3)
  300   CONTINUE
      DO 310 I = 1,NMAP
        IF (ARRAY(I,3).EQ.0.) GO TO 310
        AVN = 10.*ALOG10(ARRAY(I,4)/NPLANE)
        ARRAY(I,4) = AVN
        IF (AVN.LT.55.) GO TO 310
        ARRAY(I,5) = 3.36E-6*10.**(.103*AVN)/(.2*10.**(.03*AVN)+1.43E-4*
     1  10.**(.08*AVN))
        ARRAY(I,5) = ARRAY(I,3)*ARRAY(I,5)/PEOPLE
        ANII = ANII+ARRAY(I,5)
  310   CONTINUE
  320 TOTAL = ANII+PNALTY+THRESH+CLOSE
      IF (IGRAD.EQ.1) RETURN
      DO 350 I = 1,NTRAJ
        SXT1(I,1) = XT1(I,1)
        SXT2(I,1) = XT2(I,1)
        SYT1(I,1) = YT1(I,1)
        SYT2(I,1) = YT2(I,1)
```

```
                  DO 330 J = 2,NSEG
                    SXCENTR(I,J-1) = XCENTR(I,J-1)
                    SYCENTR(I,J-1) = YCENTR(I,J-1)
                    SXT1(I,J) = XT1(I,J)
                    SYT1(I,J) = YT1(I,J)
                    SXT2(I,J) = XT2(I,J)
                    SYT2(I,J) = YT2(I,J)
                    SANGLE(I,J-1) = ANGLE(I,J-1)
          330       CONTINUE
                  NPOSITS(I) = NPOSIT(I)
                  IPOST = NPOSITS(I)
                  DO 340 J = 1,IPOST
                    SPOSIT(I,J,1) = POSIT(I,J,1)
                    SPOSIT(I,J,2) = POSIT(I,J,2)
                    SPOSIT(I,J,3) = POSIT(I,J,3)
          340       CONTINUE
          350     CONTINUE
                RETURN
                END
```

450008 CM STORAGE USED 10.132 SECONDS

```
SUBROUTINE COST1       73/172  TS                        FTN 4.7+485       80/04/24. 10.20.35

                       SUBROUTINE COST1 (IGRAD,N,F,X,ANII,PNALTY,CLOSE,THRESH)
                       COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ARRAY(576,9),SPOSIT(3,100,3
                      1),ZO(3),ZF(3),NPOSITS(3)
                       DIMENSION X(N,1)
                       DO 10 I = 1,NTRAJ
                         NSEG1 = NSEG-1
                         DO 10 J = 1,NSEG1
                         DO 10 K = 1,2
                         L = (I-1)*2*(NSEG-1)+(J-1)*2+K
                         IF (K.EQ.1) XM(I,J+1) = X(L,1)
                         IF (K.EQ.2) YM(I,J+1) = X(L,1)
                    10   CONTINUE
                       CALL COST (IGRAD,F,ANII,PNALTY,CLOSE,THRESH)
                       RETURN
                       END

41000B CM STORAGE USED          .187 SECONDS
```

```
FUNCTION AACOS       73/172  TS                                FTN 4.7+485        80/04/24. 10.20.35

      FUNCTION AACOS (A,B,C)
      X = (A**2+B**2-C**2)/(2.*A*B)
      IF (ABS(X).LT.1.1) GO TO 10
      WRITE (6,9010) X,A,B,C
      STOP
   10 IF (X.GT.1.) X = 1.
      IF (X.LT.-1.) X = -1.
      AACOS = ACOS(X)
      RETURN
C
 9010 FORMAT (29H TROUBLE IN AACOS, X,A,B,C = ,4(1PE16.9,3X))
      END

41030B CM STORAGE USED          .095 SECONDS
```

```
      SUBROUTINE MONIT (IT,X,N,F,ANII,PNALTY,CLOSE,THRESH)
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ARRAY(576,9),SPOSIT(3,100,3
     1),Z0(3),ZF(3),NPOSITS(3)
      COMMON /PRINT/ SXCENTR(3,3),SYCENTR(3,3),SXT1(3,4),SYT1(3,4),SXT2(
     13,4),SYT2(3,4),SANGLE(3,3)
      DIMENSION X(N,1)
      DO 10 I = 1,NTRAJ
        NSEG1 = NSEG-1
        DO 10 J = 1,NSEG1
        DO 10 K = 1,2
        L = (I-1)*2*(NSEG-1)+(J-1)*2+K
        IF (K.EQ.1) XM(I,J+1) = X(L,1)
        IF (K.EQ.2) YM(I,J+1) = X(L,1)
   10   CONTINUE
      WRITE (6,9010) IT,F,ANII,PNALTY,CLOSE,THRESH
      DO 30 I = 1,NTRAJ
        WRITE (6,9020) I,XM(I,1),YM(I,1)
        PII = 1./ATAN(1.)/4.
        DO 20 J = 2,NSEG
          WRITE (6,9030) XM(I,J),YM(I,J),SXCENTR(I,J-1),SYCENTR(I,J-1),S
     1    XT1(I,J),SYT1(I,J),SXT2(I,J),SYT2(I,J),SANGLE(I,J-1)*PII*180.
   20     CONTINUE
        WRITE (6,9040) XM(I,NSEG+1),YM(I,NSEG+1)
        NPOSITI = NPOSITS(I)
        WRITE (8,9050) ((SPOSIT(I,J,K),K=1,3),J=1,NPOSITI)
   30   CONTINUE
      RETURN
C
 9010 FORMAT (1H1,9X,11HITERATION: ,I2,/,12X,12HTOTAL COST: ,1PE16.9,/,1
     12X,17HANNOYANCE (NII): ,1PE16.9,/,12X,31HPENALTY ON DYNAMIC CONSTR
     2AINT: ,1PE16.9,/,12X,34HPENALTY ON SEPARATING CONSTRAINT: ,1PE16.9
     3,/,12X,28HPENALTY ON THRESHOLD NOISE: ,1PE16.9,/)
 9020 FORMAT (12X,16HFLIGHT PATH NO: ,I1,/,14X,10HCORNER PT.,14X,16HCENT
     1ER OF CIRCLE,8X,15HTANGENTIAL PTS.,33X,13HANGLE(DEGREE),/,14X,1H(,
     2F8.1,1H,,F8.1,1H))
 9030 FORMAT (14X,4(1H(,F8.1,1H,,F8.1,1H),5X),F8.1)
 9040 FORMAT (14X,1H(,F8.1,1H,,F8.1,1H))
 9050 FORMAT (3(F8.1,2X))
      END
```

41C00B CM STORAGE USED .527 SECONDS

ORIGINAL PAGE IS
OF POOR QUALITY

```
SUBROUTINE RESULT     73/172  TS                              FTN 4.7+485        80/04/24. 10.20.35

      SUBROUTINE RESULT (IT,X,N,F,ANII,PNALTY,CLOSE,THRESH)
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ARRAY(576,9),SPOSIT(3,100,3
     1),ZO(3),ZF(3),NPOSITS(3)
      COMMON /PRINT/ SXCENTR(3,3),SYCENTR(3,3),SXT1(3,4),SYT1(3,4),SXT2(
     13,4),SYT2(3,4),SANGLE(3,3)
      DIMENSION X(N,1)
      DO 10 I = 1,NTRAJ
        NSEG1 = NSEG-1
        DO 10 J = 1,NSEG1
        DO 10 K = 1,2
        L = (I-1)*2*(NSEG-1)+(J-1)*2+K
        IF (K.EQ.1) XM(I,J+1) = X(L,1)
        IF (K.EQ.2) YM(I,J+1) = X(L,1)
   10   CONTINUE
      WRITE (6,9010) IT,F,ANII,PNALTY,CLOSE,THRESH
      DO 30 I = 1,NTRAJ
        WRITE (6,9020) I,XM(I,1),YM(I,1)
        PII = 1./ATAN(1.)/4.
        DO 20 J = 2,NSEG
          WRITE (6,9030) XM(I,J),YM(I,J),SXCENTR(I,J-1),SYCENTR(I,J-1),S
     1    XT1(I,J),SYT1(I,J),SXT2(I,J),SYT2(I,J),SANGLE(I,J-1)*PII*180.
   20     CONTINUE
        WRITE (6,9040) XM(I,NSEG+1),YM(I,NSEG+1)
        WRITE (6,9050)
        NPOSITI = NPOSITS(I)
        WRITE (6,9060) ((SPOSIT(I,J,K),K=1,3),J=1,NPOSITI)
        WRITE (6,9070) ((SPOSIT(I,J,K),K=1,3),J=1,NPOSITI)
   30   CONTINUE
      WRITE (9,9080) ((ARRAY(I,J),J=4,5),I=1,NMAP)
      RETURN
C
 9010 FORMAT (1H1,9X,11HITERATION: ,I2,/,12X,12HTOTAL COST: ,1PE16.9,/,1
     12X,17HANNOYANCE (NII): ,1PE16.9,/,12X,31HPENALTY ON DYNAMIC CONSTR
     2AINT: ,1PE16.9,/,12X,34HPENALTY ON SEPARATING CONSTRAINT: ,1PE16.9
     3,/,12X,28HPENALTY ON THRESHOLD NOISE: ,1PE16.9,/)
 9020 FORMAT (12X,16HFLIGHT PATH NO: ,I1,/,14X,10HCORNER PT.,14X,16HCENT
     1ER OF CIRCLE,8X,15HTANGENTIAL PTS.,33X,13HANGLE(DEGREE),/,14X,1H(,
     2F8.1,1H,,F8.1,1H))
 9030 FORMAT (14X,4(1H(,F8.1,1H,,F8.1,1H),5X),F8.1)
 9040 FORMAT (14X,1H(,F8.1,1H,,F8.1,1H))
 9050 FORMAT (//,12X,36HFLIGHT TRAJECTORY X, Y, Z, IN METERS,/,19X,1HX,1
     14X,1HY,14X,1HZ,/)
 9060 FORMAT (14X,1PE10.3,5X,E10.3,5X,E10.3,5X)
 9070 FORMAT (3(F8.1,2X))
 9080 FORMAT (2(1PE10.3,1X))
      END

41000B CM STORAGE USED          .643 SECONDS
```

```
      SUBROUTINE CROSOVR (NPOSIT,POSIT,FX)
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ARRAY(576,9),SPOSIT(3,100,3
     1),Z0(3),ZF(3),NPOSITS(3)
      COMMON /CROSS/ XBEGIN,XFINAL,YDIS,CWEIT1,CWEIT2
      DIMENSION CROSS(7,10), POSIT(3,100,3), NPOSIT(3)
      YINTRP(X,X1,Y1,X2,Y2,X3,Y3,X4,Y4) = (X-X2)*(X-X3)*(X-X4)*Y1/(X1-X2
     1)/(X1-X3)/(X1-X4)+(X-X1)*(X-X3)*(X-X4)*Y2/(X2-X1)/(X2-X3)/(X2-X4)+
     2(X-X1)*(X-X2)*(X-X4)*Y3/(X3-X1)/(X3-X2)/(X3-X4)+(X-X1)*(X-X2)*(X-X
     33)*Y4/(X4-X1)/(X4-X2)/(X4-X3)
      FX = 0.
      SAMPLE = (XBEGIN-XFINAL)/11.
      CROSS(1,1) = XBEGIN-SAMPLE
      DO 10 I = 2,10
        CROSS(1,I) = CROSS(1,I-1)-SAMPLE
   10   CONTINUE
      DO 70 I = 1,NTRAJ
        DO 60 J = 1,10
          NP1 = NPOSIT(I)-1
          DO 40 K = 1,NP1
            SIGN = (POSIT(I,K,1)-CROSS(1,J))*(POSIT(I,K+1,1)-CROSS(1,J))
            IF (SIGN.GE.0.0) GO TO 20
            KK = K
            GO TO 50
   20       IF (SIGN.GT.0.0) GO TO 40
            IF (POSIT(I,K,1).NE.CROSS(1,J)) GO TO 30
            CROSS(1+I,J) = POSIT(I,K,2)
            GO TO 60
   30       CROSS(1+I,J) = POSIT(I,K+1,2)
            GO TO 60
   40       CONTINUE
   50     IF (KK.EQ.1) CROSS(1+1,J) = YINTRP(CROSS(1,J),POSIT(I,1,1),POS
     1    IT(I,1,2),POSIT(I,2,1),POSIT(I,2,2),POSIT(I,3,1),POSIT(I,3,2),
     2    POSIT(I,4,1),POSIT(I,4,2))
          IF (KK.NE.1.AND.KK.NE.NP1) CROSS(1+I,J) = YINTRP(CROSS(1,J),PO
     1    SIT(I,KK-1,1),POSIT(I,KK-1,2),POSIT(I,KK,1),POSIT(I,KK,2),POSI
     2    T(I,KK+1,1),POSIT(I,KK+1,2),POSIT(I,KK+2,1),POSIT(I,KK+2,2))
          IF (KK.EQ.NP1) CROSS(1+I,J) = YINTRP(CROSS(1,J),POSIT(I,KK-2,1
     1    ),POSIT(I,KK-2,2),POSIT(I,KK-1,1),POSIT(I,KK-1,2),POSIT(I,KK,1
     2    ),POSIT(I,KK,2),POSIT(I,KK+1,1),POSIT(I,KK+1,2))
   60     CONTINUE
   70   CONTINUE
C
C ................................................................
C .                                                              .
C .   NOW TEST THE NEARNESS OF OR THE CROSSOVER BETWEEN TRAJECTORIES  .
C .   TYPICAL VALUE CWEIT1 = CWEIT2 = 0.03125                    . .
C .                                                              .
C ................................................................
C
      NADD1 = NTRAJ+1
      DO 130 I = 2,NTRAJ
        DO 120 J = 3,NADD1
          IF (I.GE.J) GO TO 120
          DO 100 K = 1,10
            DIS1 = CROSS(I,K)-CROSS(J,K)
            IF (ABS(DIS1).GE.YDIS) GO TO 80
            FX = FX+CWEIT1*(YDIS-DIS1)**2
```

```
      60     IF (K.EQ.10) GO TO 100
             DIS2 = CROSS(I,K+1)-CROSS(J,K+1)
             IF ((DIS1*DIS2).GE.0.) GO TO 100
             I1 = 1
      90     IF (K+1+I1.GT.10) GO TO 110
             DIS3 = CROSS(I,K+1+I1)-CROSS(J,K+1+I1)
             IF (DIS2*DIS3.LE.0.) GO TO 110
             IF (ABS(DIS2).LT.ABS(DIS3)) DIS2 = DIS3
             I1 = I1+1
             GO TO 90
     100     CONTINUE
           GO TO 120
     110   FX = FX+CWEIT2*DIS3**2
     120   CONTINUE
     130 CONTINUE
       RETURN
       END
```

450008 CM STORAGE USED   4.256 SECONDS

```
      SUBROUTINE QNEWTON (MAXIT,STOPCHG,N,XNOW,DELTAX)
C
C ...................................................................
C .                                                                 .
C .   THIS OPTIMIZATION EMPLOYS SELF-SCALING, RESTARTING,          .
C .   QUASI-NEWTON METHOD.                                          .
C .   REFERENCE:  D.G. LUENBERGER INTRO. TO LINEAR AND NONLINEAR    .
C .   PROGRAMMING; P.204, SEC.9.5                                   .
C .   MAXIT:  MAXIMUM NUMBER OF ITERATIONS ALLOWED                  .
C .   STOPCHG:  STOP IF PERCENTAGE CHANGE IN SUCCESSIVE COSTS IS    .
C .   LESS THAN THIS VALUE                                          .
C .   N:  DIMENSION OF THE UNKNOWN X                                .
C .   XNOW:  PRESENT OR INITIAL VALUE OF UNKNOWN X                  .
C .                                                                 .
C ...................................................................
C
      DIMENSION XNOW(N,1), DELTAX(N)
      DIMENSION GNOW(30,1), GNEXT(30,1), P(30,1), Q(30,1), PQ(1,1)
      DIMENSION QSQ(1,1), PP(30,30), SQQS(30,30), S(30,30)
      DIMENSION XTEMP(30,1), PT(1,30), QS(1,30), SQ(1,30), QT(1,30)
      DIMENSION SQQ(30,30)
      DIMENSION D(30,1)
      IT = 0
      GO TO 40
   10 FNOW = FSMALL
      DO 20 I = 1,N
        XNOW(I,1) = XTEMP(I,1)
   20   CONTINUE
      IT = IT+1
      IF (IT.LT.MAXIT) GO TO 40
      IF (PNALTY.GT.0.) GO TO 30
      CALL RESULT (IT,XNOW,N,FNOW,ANII,PNALTY,CLOSE,THRESH)
      RETURN
   30 WRITE (6,9010) PNALTY
      RETURN
   40 CALL COST1 (0,N,FNOW,XNOW,ANII,PNALTY,CLOSE,THRESH)
      IF (PNALTY.GT.0.) GO TO 50
      CALL MONIT (IT,XNOW,N,FNOW,ANII,PNALTY,CLOSE,THRESH)
      IF (MAXIT.EQ.0) STOP
      GO TO 60
   50 WRITE (6,9020) PNALTY
C
C ...................................................................
C .                                                                 .
C .   STEP 1:  SET S = IDENTITY MATRIX AND CALCULATE GRADIENT G     .
C .                                                                 .
C ...................................................................
C
   60 DO 70 I = 1,N
        DO 70 J = 1,N
        S(I,J) = 0.
        IF (I.EQ.J) S(I,J) = 1.
   70   CONTINUE
      CALL FGRAD (N,FNOW,XNOW,GNOW,DELTAX)
C
C ...................................................................
C .                                                                 .
```

```
      C *   STEP 2:  SET D = -SG                                   *
      C *
      C ***************************************************************
      C
        80 CALL MPLY (N,N,1,S,GNOW,D,30,30,1)
           DO 90 I = 1,N
             D(I,1) = -D(I,1)
        90   CONTINUE
      C
      C ***************************************************************
      C *                                                             *
      C *   STEP 3:  MOST IMPORTANT.                                  *
      C *     LINE SEARCH ALONG D TO FIND AFA THAT SATISFIES PQ>0     *
      C *     IF COST FUNCTION IS ANALYTIC, SLIGHT MODIFICATION IS    *
      C *     NEEDED (SIMPLIFICATION)(OPTIONAL)                       *
      C *                                                             *
      C ***************************************************************
      C
           K = 0
       100 K = K+1
           CALL LINESCH (K,N,FNOW,XNOW,D,AFA2,XTEMP,FSMALL,ANII,PNALTY,CLOSE,
          1THRESH), RETURNS (10,110)
       110 CALL FGRAD (N,FSMALL,XTEMP,GNEXT,DELTAX)
           DO 120 I = 1,N
             Q(I,1) = GNEXT(I,1)-GNOW(I,1)
             P(I,1) = AFA2*D(I,1)
       120   CONTINUE
           CALL TRNSPOS (30,1,P,PT)
           CALL MPLY (1,N,1,PT,Q,PQ,1,30,1)
           IF (PQ(1,1).GT.0.) GO TO 140
           IF (K.GT.4) GO TO 10
           FNOW = FSMALL
           DO 130 I = 1,N
             XNOW(I,1) = XTEMP(I,1)
       130   CONTINUE
           GO TO 100
       140 DO 150 I = 1,N
             XNOW(I,1) = XTEMP(I,1)
       150   CONTINUE
           IT = IT+1
           PRCNT = ABS((FSMALL-FNOW)/FNOW)
           IF (PRCNT.GE.STOPCHG) GO TO 170
           WRITE (6,9030) PRCNT,STOPCHG
           IF (PNALTY.GT.0.) GO TO 160
           CALL RESULT (IT,XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH)
      C
           RETURN
       160 WRITE (6,9010) PNALTY
           RETURN
       170 IF (IT.LT.MAXIT) GO TO 190
           WRITE (6,9040)
           IF (PNALTY.GT.0.) GO TO 180
           CALL RESULT (IT,XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH)
      C
           RETURN
       180 WRITE (6,9010) PNALTY
           RETURN
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
      190 IF (PNALTY.GT.0.) GO TO 200
          CALL MONIT (IT,XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH)
          GO TO 210
      200 WRITE (6,9020) PNALTY
      210 CONTINUE
C
C ..................................................................
C .                                                                .
C .   STEP 4:  IF #IT# IS INTEGER MULTIPLE OF N GO TO STEP 1;      .
C .           IF NOT, UPDATE S                                     .
C .     IS #IT# INTEGER MULTIPLE OF N+                             .
C .                                                                .
C ....,  ...........................................................
C
          IF ((FLOAT(IT)/FLOAT(N)).NE.FLOAT(IT/N)) GO TO 220
          FNOW = FSMALL
          GO TO 60
C
C ..................................................................
C .                                                                .
C .     UPDATE MATRIX S; GO TO STEP 2                              .
C .                                                                .
C ..................................................................
C
      220 CALL MPLY (N,N,1,S,Q,SQ,30,30,1)
          CALL TRNSPOS (30,1,Q,QT)
          CALL MPLY (N,1,N,SQ,QT,SQQ,30,1,30)
          CALL MPLY (N,N,N,SQQ,S,SQQS,30,30,30)
          CALL MPLY (1,N,N,QT,S,QS,1,30,30)
          CALL MPLY (1,N,1,QS,Q,QSQ,1,30,1)
          CALL MPLY (N,1,N,P,PT,PP,30,1,30)
          DO 230 I = 1,N
            DO 230 J = 1,N
            S(I,J) = (S(I,J)-SQQS(I,J)/QSQ(1,1))*(PQ(1,1)/QSQ(1,1))+PP(I,J)/
         1  PQ(1,1)
      230   CONTINUE
          FNOW = FSMALL
          DO 240 I = 1,N
            GNOW(I,1) = GNEXT(I,1)
      240   CONTINUE
          GO TO 80
C
     9010 FORMAT (5X,40HIN RESULT: DYNAMIC CONSTRAINTS VIOLATION,/,10X,9HPEN
         1ALTY =,1PE16.9)
     9020 FORMAT (5X,39HIN MONIT: DYNAMIC CONSTRAINTS VIOLATION,/,10X,9HPENA
         1LTY =,1PE16.9)
     9030 FORMAT (2X,37HPERCENTAGE CHANGE IN SUCCESSIVE COSTS,1PE10.3,25H LE
         1SS THAN STOP CRITERION,1PE10.3)
     9040 FORMAT (2X,29HMAXIMUM ITERATION SET REACHED)
          END

410008 CM STORAGE USED          1.246 SECONDS
```

```
      SUBROUTINE MPLY (L,M,N,A,B,C,LDEC,MDEC,NDEC)
C
C *****************************************************************
C *                                                               *
C *   CALCULATE MATRIX MULTIPLICATION C = AB                      *
C *                                                               *
C *****************************************************************
C
      DIMENSION A(LDEC,MDEC), B(MDEC,NDEC), C(LDEC,NDEC)
      DO 10 I = 1,L
        DO 10 J = 1,N
        C(I,J) = 0.
        DO 10 K = 1,M
        C(I,J) = C(I,J)+A(I,K)*B(K,J)
   10   CONTINUE
      RETURN
      END
```

41000B CM STORAGE USED .145 SECONDS

```
      SUBROUTINE TRNSPOS (M,N,A,B)
C
C ..................................................................
C .                                                                .
C .   TRANSPOSE OF MATRIX A IS RETURNED IN MATRIX B                .
C .                                                                .
C ..................................................................
C
      DIMENSION A(M,N), B(N,M)
      DO 10 I = 1,M
        DO 10 J = 1,N
        B(J,I) = A(I,J)
   10   CONTINUE
      RETURN
      END
```

41000B CM STORAGE USED .090 SECONDS

ORIGINAL PAGE IS
OF POOR QUALITY

```
SUBROUTINE FGRAD       73/172  TS                     FTN 4.7+485        80/04/24. 10.20.35

                 SUBROUTINE FGRAD (N,F,X,G,DELTAX)
      C
      C ..............................................................
      C .                                                            .
      C .   CALCULATE GRADIENT OF COST F WITH RESPECT TO UNKNOWN X   .
      C .                                                            .
      C ..............................................................
      C
                 DIMENSION X(N,1), G(30,1), DELTAX(30)
                 DO 10 I = 1,N
                   X(I,1) = X(I,1)+DELTAX(I)
                   CALL COST1 (1,N,FF,X,ANII,PNALTY,CLOSE,THRESH)
                   G(I,1) = (FF-F)/DELTAX(I)
                   X(I,1) = X(I,1)-DELTAX(I)
            10     CONTINUE
                 RETURN
                 END

410008 CM STORAGE USED          .137 SECONDS
```

```
      SUBROUTINE GAFA (N,F,DAFA,G,X,D)
C
C ..................................................................
C .                                                                .
C .   CALCULATE GRADIENT OF F WITH RESPECT TO AFA                  .
C .                                                                .
C ..................................................................
C
      DIMENSION X(N,1), D(30,1), XTEMP(30,1)
      DO 10 I = 1,N
        XTEMP(I,1) = X(I,1)+DAFA*D(I,1)
   10   CONTINUE
      CALL COST1 (1,N,FF,XTEMP,1,ANII,PNALTY,CLOSE,FX)
      G = (FF-F)/DAFA
      RETURN
      END
```

41000B CM STORAGE USED .104 SECONDS

```
      SUBROUTINE ERROR (K)
      WRITE (6,10) K
      STOP
C
   10 FORMAT (1X,6HAFTER ,I1,33H TIMES THROUGH LINE SEARCH, STILL,37H CA
     1NNOT FIND AFA WHICH SATISFIES PK>0)
      END
```

410008 CM STORAGE USED .036 SECONDS

```
      SUBROUTINE LINESCH (K,N,FNOW,XNOW,D,AFA2,XTEMP,FSMALL,ANII,PNALTY,
     1CLOSE,THRESH), RETURNS (M,NN)
      DIMENSION X(30,1), D(30,1), XTEMP(30,1), XNOW(N,1)
      DIMENSION X1(30,1), X2(30,1), X3(30,1), X4(30,1), AFA(4), FX(3)
C
C ..............................................................
C .                                                            .
C .   CUBIC FIT BY INITIALLY LETTING AFA = 0 AND DAFA = .01    .
C .                                                            .
C ..............................................................
C
      IF (K.LT.-100) GO TO 60
      DMAX = D(1,1)
      DO 10 I = 1,N
        IF (DMAX.LT.ABS(D(I,1))) DMAX = ABS(D(I,1))
   10   CONTINUE
      IF (DMAX.EQ.0.) CALL CHECK (1,N,FNOW,XNOW,D,AFA0,FAFA0,GAFA0,AFA1,
     1FAFA1,GAFA1,U1,UU1,U2)
      DAFA = 10./DMAX
      DAFA = 0.01
      AFA0 = 0.
      FAFA0 = FNOW
      CALL GAFA (N,FAFA0,DAFA,GAFA0,XNOW,D)
      AFA1 = 100./DMAX
      DO 20 I = 1,N
        XTEMP(I,1) = XNOW(I,1)+AFA1*D(I,1)
   20   CONTINUE
      CALL COST1 (0,N,FAFA1,XTEMP,ANII,PNALTY,CLOSE,THRESH)
   30 CALL GAFA (N,FAFA1,DAFA,GAFA1,XTEMP,D)
      IF (ABS((AFA0-AFA1)/AFA1).LT..001) CALL CHECK (2,N,FNOW,XNOW,D,AFA
     10,FAFA0,GAFA0,AFA1,FAFA1,GAFA1,U1,UU1,U2)
      U1 = GAFA0+GAFA1-3.*(FAFA0-FAFA1)/(AFA0-AFA1)
      UU1 = U1**2-GAFA0*GAFA1
      IF (UU1.LE.0.) U2 = 0.
      IF (UU1.GT.0.) U2 = SQRT(UU1)
      IF (GAFA1-GAFA0+2.*U2.EQ.0.) CALL CHECK (3,N,FNOW,XNOW,D,AFA0,FAFA
     10,GAFA0,AFA1,FAFA1,GAFA1,U1,UU1,U2)
      AFA2 = AFA1-(AFA1-AFA0)*(GAFA1+U2-U1)/(GAFA1-GAFA0+2.*U2)
      DO 40 I = 1,N
        XTEMP(I,1) = XNOW(I,1)+AFA2*D(I,1)
   40   CONTINUE
      CALL COST1 (0,N,FSMALL,XTEMP,ANII,PNALTY,CLOSE,THRESH)
      IF (FSMALL.GE.FNOW) GO TO 50
      CALL COST1 (1,N,FSMALL,XTEMP,ANII,PNALTY,CLOSE,THRESH)
      RETURN NN
   50 AFA0 = AFA1
      FAFA0 = FAFA1
      GAFA0 = GAFA1
      AFA1 = AFA2
      FAFA1 = FSMALL
      IF (AFA2.EQ.0.) DAFA = 0.01
      IF (AFA2.NE.0.) DAFA = 0.01*AFA2
      GO TO 30
C
C ..............................................................
C .                                                            .
C .   COGGIN ALGORITHM (QUADRATIC FIT)                         .
```

```
      C *                                                              *
      C ****************************************************************
      C
   60 DMAX = D(1,1)
      DO 70 I = 1,N
        X1(I,1) = XNOW(I,1)
        IF (DMAX.LT.D(I,1)) DMAX = D(I,1)
   70   CONTINUE
      DAFA = 500./DMAX
      F1 = FNOW
      SIGN = 1.
      STEP = 1.
      DO 80 I = 1,N
        X2(I,1) = XNOW(I,1)+DAFA*D(I,1)
   80   CONTINUE
      CALL COSTI ([illegible],Y2,ANII,PNALTY,CLOSE,THRESH)
   90 IF (F2.G[illegible] 110
      STEP = 2.
      DO 100 I = 1,N
        X1(I,1) = X2(I,1)
        X2(I,1) = X2(I,1)+2.*DAFA*D(I,1)*SIGN
  100   CONTINUE
      F1 = F2
      CALL COSTI (1,N,F2,X2,ANII,PNALTY,CLOSE,THRESH)
      GO TO 90
  110 IF (STEP.NE.1.) GO TO 130
      STEP = 2.
      SIGN = -1.
      DO 120 I = 1,N
        X2(I,1) = XNOW(I,1)+DAFA*D(I,1)*SIGN
  120   CONTINUE
      CALL COSTI (1,N,F2,X2,ANII,PNALTY,CLOSE,THRESH)
      GO TO 90
  130 DO 140 I = 1,N
        X3(I,1) = (X1(I,1)+X2(I,1))/2.
  140   CONTINUE
      CALL COSTI (1,N,F3,X3,ANII,PNALTY,CLOSE,THRESH)
      AFA(1) = (X1(1,1)-XNOW(1,1))/D(1,1)
      AFA(2) = (X2(1,1)-XNOW(1,1))/D(1,1)
      AFA(3) = (X3(1,1)-XNOW(1,1))/D(1,1)
  150 AFA(4) = 0.5*(((AFA(2)**2-AFA(3)**2)*F1+(AFA(3)**2-AFA(1)**2)*F2+(
     1AFA(1)**2-AFA(2)**2)*F3)/((AFA(2)-AFA(3))*F1+(AFA(3)-AFA(1))*F2+(A
     2FA(1)-AFA(2))*F3))
      DO 160 I = 1,N
        X4(I,1) = XNOW(I,1)+AFA(4)*D(I,1)
  160   CONTINUE
      CALL COSTI (1,N,F4,X4,ANII,PNALTY,CLOSE,THRESH)
      FX(1) = F1
      FX(2) = F2
      FX(3) = F3
      FMIN = FX(1)
      MIN = 1
      FMAX = FX(1)
      MAX = 1
      DO 180 I = 2,3
        IF (FMIN.LT.FX(I)) GO TO 170
        FMIN = FX(I)
```

```
                 MIN = I
        170      IF (FMAX.GT.FX(I)) GO TO 160
                 FMAX = FX(I)
                 MAX = I
        160      CONTINUE
               DO 190 I = 1,N
                 XCOMP = XNOW(I,1)+AFA(MIN)*D(I,1)
                 IF (ABS(XCOMP-X4(I,1)).LE.10.) GO TO 200
                 XTEMP(I,1) = XCOMP
        190      CONTINUE
               CALL COST1 (O,N,FSMALL,XTEMP,ANII,PNALTY,CLOSE,THRESH)
               RETURN NN
        200    AFA(MAX) = AFA(4)
               GO TO 150
               END
```

41030B CM STORAGE USED 2.336 SECONDS

```
      SUBROUTINE CHECK (ICHECK,N,FNOW,XNOW,D,AFAO,FAFAO,GAFAO,AFA1,FAFA111680
     1,GAFA1,U1,UU1,U2)                                                 11690
      DIMENSION XNOW(N,1), D(30,1)                                      11700
      WRITE (6,10) FNOW,N,(I,XNOW(I,1),D(I,1),I=1,N)                    11710
      IF (ICHECK.EQ.1) STOP                                             11720
      WRITE (6,20) AFAO,FAFAO,GAFAO,AFA1,FAFA1,GAFA1                    11730
      IF (ICHECK.EQ.2) STOP                                             11740
      WRITE (6,30) U1,UU1,U2                                            11750
      STOP                                                              11760
C                                                                       11770
   10 FORMAT (20X,44HTHIS IS SUBROUTINE CHECK WHICH GIVES ALL THE,35H IN11780
     1FORMATION IN SUBROUTINE LINESCH.,//,30X,7HFNOW = ,1PE16.9,//,37X,411790
     2HXNOW,17X,1HD,//,=(25X,I2,3X,1PE16.9,4X,1PE16.9))                 11800
   20 FORMAT (//,20X,7HAFAO = ,1PE16.9,5X,8HFAFAO = ,1PE16.9,5X,8HGAFAO 11810
     1= ,1PE16.9,/,20X,7HAFA1 = ,1PE16.9,5X,8HFAFA1 = ,1PE16.9,5X,8HGAFA11820
     21 = ,1PE16.9)                                                     11830
   30 FORMAT (//,20X,5HU1 = ,1E16.9,5X,6HUU1 = ,1PE16.9,5X,5HU2 = ,1PE1611840
     1.9)                                                               11850
      END                                                               11860
```

410008 CM STORAGE USED .143 SECONDS

ORIGINAL PAGE IS
OF POOR QUALITY